



ПРОБЛЕМИ ПРОГРАМУВАННЯ

науковий журнал

№ 4

жовтень - грудень

2024

Заснований у березні 1999 р.

ЗМІСТ

Штучний інтелект

Грішанова І.Ю., Рогушина Ю.В. Технологія використання машинного навчання для побудови композиційного веб-сервісу 3

Бердник Ю.М., Скотаренко А.О. До проблеми розпізнавання об'єктів на пристроях з обмеженими ресурсами 14

Програмна інженерія – теоретичні методи

Шкільняк С.С. Першопорядкові логіки з частковими предикатами для перевірки означеності змінної 23

Прикладне програмне забезпечення та інформаційні системи

Дорошенко А.Ю., Гайдукевич Я.О., Гайдукевич В.О., Жиренков О.С. Клієнто – центричний технологічний стек для прогнозу погоди та якості повітря 34

Єфремов М.С., Крак Ю.В. Програмне рішення для зберігання та аналізу електрокардіограм 43

Кузьміна К.І., Сьомик Т.М., Андон А.П. Розробка модуля соціопсихологічної підтримки для сучасних ІТ, як засобу підвищення ефективності діяльності особистості і колективу при збереженні їх біосоціального здоров'я 51

Комп'ютерне моделювання

Grygoryan R.D., Degoda A.B., Lyudovyk T.V., Yurchak O.I. Simulating of human physiological supersystems: integrative function of organs supporting cell life 77

Програмні системи захисту інформації

Шуклін Г.В., Барабаш О.В., Гребенніков А.Б. Властивості функціональної стійкості інтелектуальних систем у процесі ухвалення рішень 89

Семантик Веб та лінгвістичні системи

Резніченко В.А., Захарова О.В. Опис процесу веб-сервісу. Таблична інтерпретація 99

Програмна інженерія виробництва програмних продуктів

Sydorov N.A. Ecosystem research in software engineering 114

Свідоцтво про державну реєстрацію КВ № 7490 від 01.07.2003

Науковий журнал "Проблеми програмування" занесений до переліку наукових фахових видань України, в яких можуть публікуватися основні результати дисертаційних робіт.



PROBLEMS OF PROGRAMMING

scientific journal

№ 4

October – December

2024

Founded in March, 1999

CONTENT

Artificial Intelligence

Grishanova I.Yu., Rogushina J.V. The technology of machine learning for a composite web service development 3

Berdnyk Yu.M., Skotarenko A.O. On the problem of object recognition on devices with limited resources 14

Software Engineering – Theoretical Methods

Shkilniak S.S. First-order logics with partial predicates for checking variable definedness 23

Applied Software and Information Systems

Doroshenko A.Yu., Haidukevych Y.O., Haidukevych V.O., Zhyrenkov O.S. User-centric technology stack for weather and air pollution forecasting 34

Yefremov M.S., Krak Yu.V. Software solution for electrocardiogram storage and analysis 43

Kuzmina K.I., Somik T.M., Andon A.P. Development of a module of sociopsychophysiological support for modern IT as a means of increasing the efficiency of individual and collective activities while preserving their biosocial health 51

Computer Modelling

Grygoryan R.D., Degoda A.B., Lyudovyk T.V., Yurchak O.I. Simulating of human physiological supersystems: integrative function of organs supporting cell life 77

Software for Secure Information

Shuklin G.V., Barabash O.V., Grebennikov A.B. Functional stability of intelligent systems in decision-making 89

Semantic Web and Linguistic Systems

Reznichenko V.A., Zakharova O.V. Description of the web service process. Tabular interpretation 99

Software Engineering for Creating Software Products

Sydorov N.A. Ecosystem research in software engineering 114

І. Гришанова, Ю. Рогушина

ТЕХНОЛОГІЯ ВИКОРИСТАННЯ МАШИННОГО НАВЧАННЯ ДЛЯ ПОБУДОВИ КОМПОЗИЦІЙНОГО ВЕБ-СЕРВІСУ

Проаналізовано алгоритми динамічного програмування та машинного навчання (на прикладі Q-learning), що використовуються для автоматичної адаптивної композиції веб-сервісів на основі оцінок якості сервісів, їхні вхідні параметри та особливості роботи. Для порівняння параметрів роботи цих алгоритмів на наборах сервісів різного обсягу розроблено програмну реалізацію.

Визначено, що розглянуті методи дозволяють знаходити оптимальний набір сервісів тільки для композиції з попередньо визначеним маршрутом фіксованої довжини. Тому виникає потреба узагальнити постановку задачі для довільного набору класів у маршруті композиції сервісів. На основі виконаного аналізу розроблено алгоритм розв'язання цієї задачі – побудови композитного сервісу з маршрутом довільної довжини методом Q-Learning, що має найкращі сумарні оцінки якості. Розроблено програмну реалізацію як цього алгоритму, так і інших алгоритмів розв'язання цієї задачі (генетичний алгоритм, жадібний пошук, динамічне програмування, SARSA), яка дозволила порівняти швидкість їх роботи та оцінки результуючого композитного сервісу на наборах даних різного обсягу.

Ключові слова: веб-сервіс, композиція сервісів, машинне навчання.

I. Grishanova, J. Rogushina

THE TECHNOLOGY OF MACHINE LEARNING FOR OF A COMPOSITE WEB SERVICE DEVELOPMENT

We analyze dynamic programming and machine learning algorithms (on example of Q-learning) used for automatic adaptive composition of web services based on service quality assessments, their input parameters and work specifics. Software implementation of these algorithms on sets of services of different volumes is developed for comparison their performance parameters.

We determine that the considered methods allow finding the optimal set of services only for composition with a predefined fixed-length route. This restriction causes a need to generalize the problem formulation for an arbitrary set of service classes in the composition route. On base of the performed analysis, we developed an algorithm that solves this problem of building a composite service with a route of arbitrary length (using the Q-Learning method), that has the best overall quality ratings. A software implementation of both this algorithm and other algorithms for solving this problem (genetic algorithm, greedy search, dynamic programming, SARSA, etc.) are developed to compare the speed of their work and the evaluation of the resulting composite service on data sets of different volumes.

Keywords: web service, composition of services, machine learning.

Вступ

У сучасному світі інформаційних технологій веб-сервіси стали важливим інструментом для надання функціональності через мережу. Веб-сервіси дозволяють розробникам об'єднувати окремі програмні компоненти для створення складних, функціональних та інтегрованих рішень. Композитні веб-сервіси відіграють ключову роль у створенні таких рішень, дозволяючи використовувати кілька сервісів у межах єдиного ро-

бочого процесу для досягнення поставлених цілей.

Композиція сервісів є активною сферою досліджень, яка застосовує розробки зі сфери машинного навчання та семантичних технологій [1], для перетворення веб-середовища на розподілену обчислювальну платформу. Ця діяльність потребує створення та дослідження відповідних концепцій, моделей, алгоритмів, платформ та інструментів.

Процес *композиції* веб-сервісів є досить складним, особливо в умовах динамічного середовища, де сервіси можуть бути тимчасово недоступними або змінювати свої характеристики, зокрема, параметри *якості обслуговування* (QoS). Крім того, потрібно швидко адаптувати систему до нових умов, обирати оптимальні сервіси, а також оптимізувати композитний сервіс за допомогою параметрів QoS.

Один із перспективних підходів до автоматичної адаптивної композиції веб-сервісів базується на використанні машинного навчання, зокрема, навчання з підкріпленням, у якому агент навчається на основі взаємодії із середовищем через отримання винагороди за свої дії.

Використання одного з основних алгоритмів навчання з підкріпленням – *Q-learning* – дозволяє агенту поступово вивчати оптимальну стратегію вибору веб-сервісів для побудови композитних сервісів: він дозволяє агенту коригувати свої рішення на основі зворотного зв'язку від середовища, не потребуючи повної моделі цього середовища.

Машинне навчання і Reinforcement Learning

Машинне навчання – це підгалузь штучного інтелекту, яка полягає у використанні алгоритмів і моделей, що дозволяють комп'ютерам здобувати знання з даних та знаходити рішення без чіткої програми. Один із важливих класів машинного навчання – *навчання з підкріпленням* (reinforcement learning, RL).

Навчання з підкріпленням вирізняється тим, що агент взаємодіє із середовищем у режимі реального часу, отримуючи зворотний зв'язок у вигляді винагороди або штрафу залежно від того, наскільки успішно його дія привела до досягнення поставленої мети.

Однією з найбільш популярних технік RL є група алгоритмів *Q-learning*, де агент будує таблицю *Q-значень* для різних станів і дій, поступово знаходячи оптимальну стратегію (*policy*) через спроби і помилки. На відміну від інших алгоритмів RL, *Q-learning* використовує прості *Q-*

функції. Існує велика кількість підтипів *Q-learning* – з одним або кількома агентами, з різними типами взаємодії [2].

Завдяки тому, що агент постійно оновлює свої знання на основі отриманих винагород, він здатний адаптуватися до змін у середовищі, що робить його ефективним інструментом для задач, де середовище є динамічним або частково відомим.

У контексті веб-сервісів навчання з підкріпленням стає особливо важливим, оскільки середовище веб-сервісів є динамічним та складним, веб-сервіси можуть змінювати свої характеристики якості (*Quality of service – QoS*) [3], або ставати недоступними. Тому агент, що навчається, повинен вміти адаптуватися і швидко підбирати нові рішення для композиції.

Для побудови композиції сервісів потрібно спочатку визначити множину можливих маршрутів *Flow* (автоматизовано або вручну, як робить більшість дослідників), що визначає послідовність виконання сервісів, а потім для таких маршрутів визначити оптимальний набір сервісів.

Навчання з підкріпленням дозволяє агенту самостійно знаходити оптимальні комбінації веб-сервісів на основі їхніх параметрів QoS, причому значення цих параметрів можуть змінюватися динамічно в процесі виконання сервісів. Підхід *Q-learning* надає можливість вирішувати ці завдання через поступове покращення стратегії складання композитного сервісу, який буде найбільш оптимальним за заданими параметрами.

Задача композиції веб-сервісів

Розглянемо задачу створення композитного сервісу, взявши за основу приклад, наданий у роботі [4], поступово змінюючи вимоги для досягнення більш адаптивних можливостей в композиції веб-сервісів.

У контексті композиції веб-сервісів, *композитний сервіс* (КС) складається з набору окремих веб-сервісів, що виконують певні завдання, які об'єднуються для досягнення конкретної мети або виконання складної операції в певній послідовності.

Основними компонентами побудови композитного сервісу є:

1. **Атомарні вебсервіси:** це окремі веб-сервіси, які можуть виконувати певні операції або надавати функціональність, наприклад, отримання даних про погоду, конвертацію одиниць виміру тощо. Кожен такий атомарний веб-сервіс має атрибути якості (QoS), такі як: *Доступність (Availability)*: Частота успішних викликів; *Час виконання (Execution Time)*: Час, необхідний для виконання запиту; *Пропускна здатність (Throughput)*: Кількість запитів, які можуть бути оброблені за одиницю часу.
2. **Класи веб-сервісів:** веб-сервіси можуть бути розподілені на *класи*, залежно від їхньої функціональності. Наприклад, один клас може містити веб-сервіси для отримання даних про погоду, інший – для конвертації одиниць виміру температури. КС складається з вибору веб-сервісів із кожного класу, які виконуються у певній послідовності.
3. **Маршрут (Flow):** КС має впорядковану послідовність виконання атомарних веб-сервісів, де кожен із сервісів виконує свою задачу, а вихід одного сервісу може бути входним для іншого. Зокрема, перший сервіс отримує дані про температуру, а другий конвертує отриману температуру в іншу одиницю виміру (наприклад, з Фаренгейта в Цельсій).
4. **Оптимізація за параметрами якості:** КС формується таким чином, щоб мінімізувати або максимізувати певні показники якості сервісу, як-от, мінімальний час виконання або максимальну доступність.

Постановка задачі

В роботі аналізуються існуючі рішення для побудови композитних сервісів, що дозволяють це робити для зафіксованого маршруту, який визначає послідовності виконання сервісів, і обрати оптимальний

набір сервісів. Оптимальність оцінюється на основі характеристик якості атомарних сервісів, що входять до складу КС.

Проблема полягає у тому, що більшість дослідників цієї задачі використовують різні форми опису маршруту та входних даних сервісів, які значно обмежують сферу застосування методів композиції, звужуючи її до фіксованого плану виконання сервісів. Це зменшує потенційну гнучкість сервіс-орієнтованої архітектури та потребує дослідження більш узагальнених початкових умов. Такий підхід потребує також дослідження методів побудови маршрутів та критеріїв їх порівняння, але ці дослідження є поза рамками даної публікації.

Щоб визначити можливості та недоліки алгоритмів вибору атомарних сервісів для побудови композитного сервісу на основі маршруту, потрібно створити програмну реалізацію різних варіантів для алгоритмів композиції сервісів та порівняти швидкість і якість їх виконання та навчання. Це дозволить визначити, які з цих алгоритмів найбільш придатні для розв'язання задачі композиції у динамічному середовищі і перспективні для подальшого удосконалення та які параметри впливають на їхню ефективність.

Методи композиції веб-сервісів з використанням QoS

Зараз існує велика кількість методів композиції веб-сервісів на основі QoS, що різняться часом виконання, структурою входних даних та кількістю ітерацій. Всі вони використовують модель Марковського Процесу Прийняття Рішення - Markov Decision Process (MDP) для ухвалення рішення щодо оптимальності кожного з окремих сервісів, що розглядаються як кандидати на виконання.

Метод динамічного програмування

Метод динамічного програмування [5] використовує алгоритми Policy Iteration, Value Iteration та Iterative Policy Evaluation [6].

Програмна реалізація цих алгоритмів¹ показала, що алгоритм Policy Iteration найшвидше знаходить оптимальні стратегії композиції веб-сервісів (з мінімальною кількістю ітерацій), тоді як Value Iteration та Iterative Policy Evaluation показують задовільні результати, але потребують більше часу.

Алгоритм Q-learning з фіксованим маршрутом композитного сервісу

Алгоритм Q-learning з фіксованим маршрутом композитного сервісу і без динамічної зміни класів сервісів та їх кількості реалізує композицію веб-сервісів за допомогою навчання RL, а саме – використовуючи Q-learning. Цей підхід може використовувати ті ж самі вхідні дані, що й попередньо розглянуті динамічні методи, але з акцентом на дослідженні простору станів без попередньо відомої функції переходів.

У програмній реалізації² алгоритм Q-learning для вибору оптимальної послідовності веб-сервісів використовує такі параметри QoS, як доступність (availability), час виконання (execution time) та пропускна здатність (throughput). КС формується через навчання агента на основі нагород, що відповідають атрибутам QoS. Агент взаємодіє з середовищем і поступово вчиться вибирати послідовність дій (тобто, виклик веб-сервісів) таким чином, щоб максимізувати сумарну винагороду. Наприклад, якщо агент вибирає сервіс із високою доступністю, але з довгочасним виконанням, він отримує меншу винагороду, ніж якщо він обере сервіс із збалансованими QoS-показниками.

Тестування показало, що алгоритм навчання з підкріпленням потребує більше

часу порівняно з динамічним програмуванням (Policy Iteration або Value Iteration), оскільки Q-learning досліджує простір станів через багаторазові ітерації. Q-таблиця після завершення навчання зберігає найкращі очікувані винагороди для кожної пари "стандія". Це дозволяє агенту обирати оптимальні дії на основі попереднього досвіду.

З отриманих результатів можна зробити висновки: динамічні алгоритми (Policy Iteration, Value Iteration): працюють швидше, оскільки не вимагають дослідження простору станів та вимагають точного знання функцій переходів між станами, тоді як Q-learning більше відповідає середовищам з невідомими функціями переходів та потребує більше ітерацій, оскільки досліджує простір станів експериментальним шляхом. Таким чином, для проблеми композиції веб-сервісів, якщо відомі функції переходів між станами, динамічні методи показують кращі результати як за часом, так і за ефективністю. Проте, Q-learning є надійним рішенням для випадків, де функції переходів невідомі або змінюються динамічно.

Отримані нами висновки співпадають з результатами інших дослідників, але їхні підходи, а саме структури даних, не дозволяють знаходити оптимальний набір сервісів для композиції з попередньо не визначеним маршрутом. Тому виникає потреба узагальнити постановку задачі для довільного набору класів у маршруті композиції сервісів.

Знаходженням оптимальної композиції сервісів для довільного маршруту

Розглянемо використання Q-learning для композиції сервісів і знаходженням оптимальної композиції із заданими наборами груп вебсервісів і набором Flow композитного сервісу для довільного маршруту, який не зафіксовано явно до початку побудови композитного сервісу.

У попередньому прикладі з використанням Q-learning для композиції веб-сервісів не було явного опису композитного сервісу або Flow, але він був заданий алгоритмічно. КС Flow має на увазі послі-

- ---
- 1 <https://github.com/TurtleIren/test-ML-methods/blob/3db8d92c050b5dc66a65914756a988b8aad51d1e/WSComposition-VI-PI-IPE-compare.py>
- 2 <https://github.com/TurtleIren/test-ML-methods/blob/55315be65031ab0144ae6beec8c55ba01abb59ea/WSCompositionQ-learn-fixed-1.py>

довність кількох окремих веб-сервісів, кожен з яких виконує частину загального завдання. У термінах коду Flow має бути представлено як послідовний вибір кількох дій (веб-сервісів) з різних класів, які спільно реалізують КС.

Ціллю алгоритму є знаходження оптимального КС із набору існуючих Flow, використовуючи алгоритм Q-learning. Після тренування алгоритм повинен обирати оптимальний Flow на основі максимальної суми винагород для кожного композитного сервісу. Такий алгоритм має розширений набір вхідних даних:

- *Класи веб-сервісів (service_classes)*: кожен клас, як і у розглянутих вище алгоритмах, містить набір веб-сервісів, згрупованих за функціональними характеристиками, із явно заданими параметрами QoS. Класи явно визначені, наприклад, клас 1 містить "сервіс_1", "сервіс_2";
- *Flow*: до вхідних даних додається опис множини маршрутів Flow, який визначає припустимі послідовності класів веб-сервісів довільної скінченої довжини, які можуть бути використані для створення композитного сервісу. Наприклад, flows = [[1, 2], [1, 3, 4]] означає, що перший Flow складається з веб-сервісу з класу 1 і веб-сервісу з класу 2, а другий Flow – з послідовності веб-сервісів з класів 1, 3 і 4;
- *Параметри QoS*: значення параметрів якості веб-сервісу (такі як доступність, час виконання та пропускну здатність).

Для реалізації поставленої задачі пропонується алгоритм композиції веб-сервісів із використанням Q-learning, який може адаптуватися до різної кількості веб-сервісів у класах і Flow, і забезпечити гнучкість у визначенні композиції (Flow) веб-сервісів.

Відповідно до моделі MDP для адаптивної композиції веб-сервісів, цей алгоритм описується через наступні терміни:

- *Стан (state)*: кожен стан представляє часткову композицію веб-сервісів.
- *Дії (actions)*: вибір веб-сервісу з певного класу для додавання в поточний Flow.
- *Нагорода (reward)*: враховує атрибути QoS (доступність, час виконання, пропускну здатність).
- *Q-learning*: алгоритм для навчання.

Цей алгоритм дозволяє обробляти адаптивні дані, а саме:

- кількість класів веб-сервісів може змінюватись;
- Flow композитного сервісу може містити різну кількість веб-сервісів на різних етапах;
- дані можуть генеруватись випадково для тестування різних сценаріїв.

Програмна реалізація цього методу³ містить явно задані дані про веб-сервіси. У масиві service_data кожен веб-сервіс має фіксовані значення для атрибутів QoS: availability, exec_time, та throughput. Однак у процесі експлуатації ці значення можуть змінюватись, і відповідно до цих змін буде змінюватись значення винагороди і відповідно оптимальний Flow. Винагорода розраховується на основі значень QoS для обраного веб-сервісу. Для кожного Flow генерується КС. Метод find_optimal_flow() знаходить оптимальний Flow на основі суми винагород. Завдяки новій структурі з явним заданням Flow можна мати різну кількість веб-сервісів на різних етапах композиції. Крім того, можна змінювати кількість епізодів для навчання та інші параметри.

Після навчання для кожного Flow виводиться набір сервісів, які були обрані

³ <https://github.com/TurtleIren/test-ML-methods/blob/068c4f1fdc0970702b448398e7e13dad4988955e/WSCCompositionQ-learn-adaptive.py>

як оптимальні саме для цього маршруту.

Наприклад:

Згенерований композитний сервіс для Flow 1: ['сервіс_1', 'сервіс_3'] із загальною винагородою: 41.233

Згенерований композитний сервіс для Flow 2: ['сервіс_1', 'сервіс_6', 'сервіс_8'] , із загальною винагородою: 48.924

Потім алгоритм обирає найкращий Flow, ґрунтуючись на сукупній винагороді за всі сервіси, що увійшли до отриманого Flow. Таким чином, програма генерує композитний веб-сервіс із найкращою сумарною оцінкою якості атомарних сервісів.

Найкращий Flow: 2, із загальною винагородою: 48.924

Програма призначена для вирішення задачі композиції веб-сервісів із метою визначення оптимального композитного сервісу на основі наданих класів веб-сервісів і можливих Flow. КС формується на основі вибору сервісів з різних класів за допомогою підходу Q-learning, який належить до методів навчання з підкріпленням (reinforcement learning).

Програма здійснює навчання на основі винагород (reward) для вибору веб-сервісів із різних класів так, щоб максимізувати загальну якість сервісів (QoS). Основні модулі програми:

1. Ініціалізація (конструктор класу):

Програма встановлює список станів та можливих дій для кожного Flow. Станами є різні етапи композитного Flow, а діями – вибраний веб-сервіс з відповідного класу.

2. Генерація станів: Метод generate_states() створює список можливих станів для кожного Flow.

3. Генерація дій: Метод generate_actions() генерує всі можливі дії для кожного стану, де кожна дія – це вибір одного веб-сервісу з певного класу.

4. Обчислення винагороди (QoS):

Метод get_reward() обчислює винагороду за вибір конкретного веб-сервісу на основі його параметрів якості. Формула для винагороди враховує такі параметри,

як доступність, час виконання і пропускну здатність:

$$\text{reward} = (\text{availability} \times K1) - (\text{execution time} \times K2) + (\text{throughput} \times K3)$$

Значення коефіцієнтів важливості параметрів K1, K2 та K3 у формулі винагороди визначають вагу кожного параметра якості обслуговування (QoS): доступність (availability), час виконання (execution time) та пропускну здатність (throughput). Ці значення обираються емпірично або за вимогами користувача (наприклад, в даній програмній реалізації обрано K1=0.4, K2=0.4 та K3=0.2). Таким чином, веб-сервіси з високою доступністю та пропускну здатністю і мінімальним часом виконання отримують вищу винагороду.

5. Вибір дії (epsilon-greedy): Для вибору дій програма використовує епсілон-жадібну (epsilon-greedy) стратегію, яка дозволяє знаходити баланс між дослідженням нових сервісів та використанням уже знайдених оптимальних рішень: з імовірністю ϵ програма обирає випадковий веб-сервіс для дослідження (exploration); з імовірністю $1-\epsilon$ програма обирає дію з найвищим Q-значенням на основі поточної Q-таблиці.

6. Оновлення Q-таблиці: Метод update_q_table() використовує стандартну формулу Q-learning для оновлення значень Q у таблиці на основі поточної винагороди та очікуваної максимальної винагороди для наступного стану.

7. Навчання: Метод train() виконує навчання протягом заданої кількості епізодів (прикладів навчання). Під час навчання система проходить через кожен Flow, вибираючи сервіси, обчислюючи винагороди та оновлюючи Q-таблицю.

8. Генерація композитного сервісу: Метод generate_composite_service()

генерує КС для кожного Flow на основі навченої Q-таблиці, обираючи сервіси з найвищим значенням Q у кожному стані.

9. *Пошук оптимального Flow*: Метод `find_optimal_flow()` обчислює загальну винагороду для кожного Flow та визначає найкращий КС на основі максимальної суми винагород.

Алгоритм адаптується до різних Flow, які представляють різні варіанти складання композитних сервісів із класів веб-сервісів. Підхід із використанням навчання з підкріпленням дозволяє знайти оптимальний КС, орієнтуючись на параметри якості сервісів (QoS), такі як доступність, час виконання та пропускну здатність.

Визначення коефіцієнтів важливості винагороди

Визначення коефіцієнтів важливості винагороди може здійснюватися кількома різними способами:

Емпіричний підхід: Значення можуть бути підбрані емпірично на основі попереднього досвіду або аналізу системи. Вони можуть бути скориговані після серії експериментів або тестувань для забезпечення правильного балансу між різними параметрами QoS.

Вимоги системи або користувача: у різних системах або для різних сервісів деякі параметри можуть бути важливішими за інші. Наприклад, у критичних системах доступність може мати найбільший пріоритет, оскільки від неї залежить надійність сервісу. Інші системи можуть вимагати мінімізації часу виконання або оптимізації пропускну здатності.

Збалансований підхід: коефіцієнти також можуть бути підбрані для збалансованого впливу на загальну винагороду. Зокрема, використання однакових коефіцієнтів для доступності та часу виконання може означати, що система хоче забезпечити стабільну роботу сервісів з хорошою доступністю і водночас підтримувати короткий час відповіді. Менша вага для пропускну здатності свідчить про те, що її важливість у цьому сценарії менша.

Для визначення цих значень, необхідно здійснити наступне:

- *Аналіз потреб системи*: Якщо деякі параметри QoS є важливішими для вашої системи, ці значення можуть бути змінені відповідно до вимог. Наприклад, якщо в задачі важливий час виконання, можна збільшити коефіцієнт для `execution time`.
- *Тестування і оптимізація*: В процесі навчання з підкріпленням здійснюються серії тестів з різними значеннями коефіцієнтів для визначення найкращих налаштувань, які забезпечують оптимальну роботу системи.
- *Моделювання користувацького досвіду*: У деяких випадках користувачі можуть задати пріоритетність параметрів, виходячи з їхніх потреб. Наприклад, у хмарних сервісах для певних користувачів важливіша доступність, аніж час виконання, тоді як для інших навпаки.

Порівняння алгоритмів композиції сервісів

Крім Q-Learning, що розглянутий вище, для композиції сервісів можуть використовуватися інші алгоритми. Розглянемо найбільш поширені з них.

Алгоритм жадібного пошуку (Greedy algorithm): на кожному кроці обирається той сервіс, який має максимальну винагороду на поточному етапі, без урахування майбутніх винагород. Такий підхід не враховує довгострокові наслідки вибору на кожному етапі. Хоча він швидкий, але не гарантує глобально оптимальних результатів, оскільки може пропустити вигідніші дії у довгостроковій перспективі.

Еволюційні алгоритми (Genetic Algorithms): можуть використовуватися для пошуку оптимальних рішень у просторі можливих Flow, але вони потребують значного обчислювального часу та надто залежні від налаштування параметрів;

Динамічне програмування: складне завдання розбивається на підзадачі і вирі-

шує їх шляхом обчислення оптимальних результатів для менших підзадач. Алгоритм зберігає проміжні рішення для уникнення повторних обчислень. Він використовує таблицю для запам'ятовування оптимальних рішень для кожного стану. Як і жадібний алгоритм, динамічне програмування шукає оптимальні рішення, але на основі глобального аналізу. Цей підхід може бути використаний для знаходження оптимальних рішень, але він також вимагає попереднього знання всіх можливих станів і дій.

SARSA: метод навчання з підкріпленням, яке, на відміну від Q-learning, оновлює значення Q-функції на основі виконаної дії та отриманої наступної дії (Q-Learning використовує максимальне Q-значення для наступної дії). SARSA [7] розглядає поточний стан, дію, винагороду, наступний стан і наступну дію, що робить його менш агресивним, але стабільнішим у змінних середовищах. Отож, SARSA є більш консервативним методом, оскільки оновлює Q-функцію відповідно до реальних дій, а не теоретично максимальних.

Щоб порівняти Q-Learning із цими методами, проведемо їх тестування на однакових наборах даних. Для проведення такого тестування ми змінюємо механізм отримання даних, а саме дані будуть братися із згенерованих csv-файлів.

Згенеруємо масив текстових даних для сценарію, де максимальна кількість сервісів у класі становить 100, максимальна довжина Flow – 100 сервісів⁴. Кожен клас міститиме довільну кількість сервісів, а Flow складатимуться з довільних класів і сервісів. Крім того, для кожного сервісу згенеруємо параметри QoS. Ми використовуємо наступні домовленості:

- Класи *вебсервісів*: У кожного класу випадкова кількість сервісів від 1 до 100. Для кожного сервісу генеруються параметри якості (QoS), такі як доступ-

ність, час виконання та пропускна здатність.

- *Flow*: Випадковим чином генеруються Flow з різних класів. Максимальна кількість класів у Flow – 100.

Файл *services.csv*⁵ містить інформацію про всі сервіси та їхні параметри QoS, а *flows.csv*⁶ містить послідовності Flow для тестування.

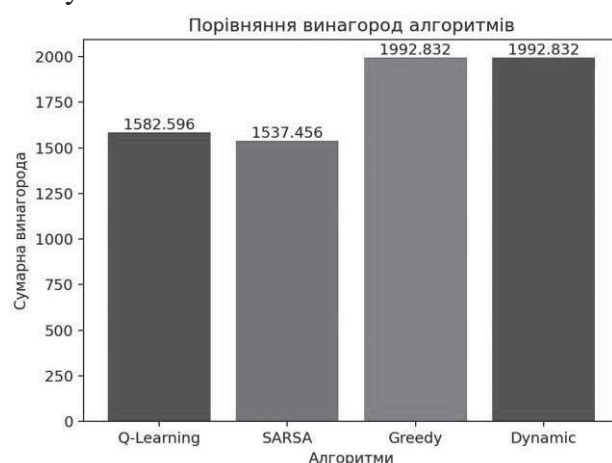


Рис. 1. Порівняння винагород знайденого композитного сервісу різними алгоритмами

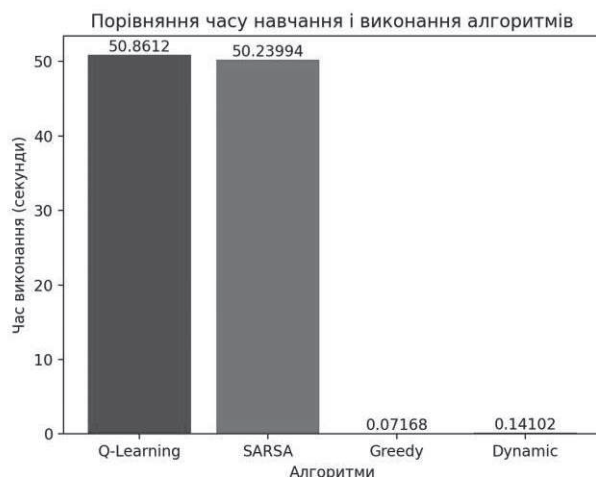


Рис. 2. Порівняння часу навчання побудови композитного сервісу.

- https://github.com/TurtleIren/test-ML-methods/blob/2bb687e16be812e74c613e2b4b919e136ca44860/services_100.csv
- https://github.com/TurtleIren/test-ML-methods/blob/2bb687e16be812e74c613e2b4b919e136ca44860/flows_100.csv

Результати порівняльного тестування⁷ показали наступне (Рис.1, Рис.2):

1. Q-Learning:

Переваги: Добре підходить для середовищ, де система потребує багато дослідження і поступового покращення на основі довгострокових вигод. Однак час виконання задовгий (50.57 секунд), а результат є субоптимальним (винагорода 1552.072), порівняно з іншими методами.

Недоліки: Час виконання значно більший порівняно з жадібним алгоритмом і динамічним програмуванням.

2. SARSA:

Переваги: Стабільніший у середовищах з динамічними змінами, оскільки враховує реальні дії. Час виконання (49.99 секунд) дещо менший, ніж у Q-Learning, а результат (винагорода 1574.544) дещо кращий.

Недоліки: Час виконання також доволі високий, і може не завжди знаходити глобально оптимальні рішення.

3. Жадібний алгоритм:

Переваги: Найшвидший алгоритм (0.07192 секунд), оскільки він просто обирає найкращу дію на кожному кроці без глибоких розрахунків. Він досягає найвищої винагороди (1992.832), що робить його ефективним для композиції веб-сервісів.

Недоліки: Може пропустити оптимальніші глобальні рішення через короткозорість, якщо деякі рішення вимагають короткострокових втрат.

4. Динамічне програмування:

Переваги: Показує другий найкращий результат після жадібного алгоритму, але також знаходить оптимальний Flow з найвищою винагородою (1992.832) і виконується дуже швидко (0.14223 секунд).

Недоліки: Використовує більше пам'яті, оскільки зберігає всі проміжні результати, але це виправдано швидкістю та результатами.

Нами також була спроба виконати поставлену задачу *методом генетичного алгоритму*. Цей метод працював довше на

невеликих наборах даних, однак на тестовому прикладі він не зміг виконати поставлену задачу. Генетичний алгоритм часто потребує багатьох ітерацій, щоб зійтися до оптимального або навіть близького до оптимального рішення. Це призводить до повільного виконання порівняно з іншими методами. Генетичний алгоритм не завжди гарантує глобальне оптимальне рішення. Через випадкові процеси він може потрапити у локальні оптимуми, і без належного налаштування може зупинитися на субоптимальних рішеннях. Генетичний алгоритм потребує налаштування багатьох гіперпараметрів: розмір популяції, ймовірність мутації, ймовірність кросовера, умови зупинки тощо. Неправильне налаштування цих параметрів може значно вплинути на продуктивність і точність алгоритму. Випадкові операції мутації та кросовера можуть призвести до менш ефективного рішення або навіть віддалити від оптимального. В свою чергу це може призвести до нестабільних результатів між різними запусками алгоритму. Отже, генетичний алгоритм має потенціал для знаходження глобальних оптимальних рішень, але для задач композиції веб-сервісів на основі QoS з великими наборами даних він менш ефективний через високу обчислювальну складність і тривалість конвергенції.

Таким чином, можна зробити висновок, що для швидкої композиції веб-сервісів на основі QoS найкраще використовувати жадібний алгоритм або динамічне програмування, оскільки вони демонструють відмінну швидкість виконання і знаходять глобально оптимальні рішення. Q-Learning та SARSA краще підходять для середовищ, де важливіша адаптація до динамічних змін і дослідження можливих рішень у довгостроковій перспективі, але вони працюють повільніше і можуть знаходити субоптимальні рішення.

Висновки

В роботі розглянуто застосування Q-learning для знаходження оптимального композитного сервісу з урахуванням даних QoS. Запропонована архітектура та структура даних є більш гнучкими та дозволя-

⁷ <https://github.com/TurtleIren/test-ML-methods/blob/30b8fd6ed80a44bc776693ac6a35e6137ca3881c/WSCComposition-compare.py>

ють підтримувати більше класів або типів сервісів і містять потенціал для подальшого удосконалення адаптивності процесу автоматизації побудови КС. Структура станів є складнішою, де кожен стан може відображати наявні вхідні та вихідні дані веб-сервісів, що дає змогу використовувати більший простір станів і більше можливих шляхів для композиції. Використання однієї спільної Q-таблиця для всіх станів і дій дозволяє гнучкіше масштабувати та обробляти складніші комбінації. Запропонована програмна реалізація дозволяє продемонструвати переваги цього рішення. Відповідно до концепції відкритої науки щодо відтворюваності результатів, створений програмний код є у відкритому доступі, і його можна отримати за посиланнями, наведеними у статті.

Здійснене тестування рішення задачі композиції веб-сервісів різними методами, з якого можна зробити висновок, що для швидкої композиції веб-сервісів на основі QoS найкраще використовувати жадібний алгоритм або динамічне програмування, оскільки вони демонструють відмінну швидкість виконання і знаходять глобально оптимальні рішення. Але ці методи не придатні для великих або невідомих просторів станів і потребують повного знання середовища. Методи Q-Learning та SARSA краще придатні для середовищ, де важливіше значення мають адаптація до динамічних змін і дослідження можливих рішень у довгостроковій перспективі, але ці методи працюють повільніше і можуть знаходити субоптимальні рішення.

Тестування програми, що базується на використанні Q-learning, демонструє його ефективність у знаходженні оптимальних композитних сервісів на різних тестових наборах даних. Найкращий композитний сервіс обирається на основі максимізації суми винагород за параметрами якості сервісів QoS. Q-learning є більш гнучким порівняно з альтернативними підходами (такими як жадібний пошук та еволюційні алгоритми), оскільки він адаптується до змін у середовищі та дозволяє оцінювати довгострокові наслідки вибору на кожному етапі.

Наступним кроком у подальших дослідженнях може бути автоматизація складання маршрутів Flows для веб-сервісів з урахуванням змінних параметрів середовища. Автоматизація дозволить не тільки динамічно адаптувати сервіси під нові умови, а й покращити якість і швидкість складання оптимальних маршрутів на основі QoS. Крім того, доцільно буде дослідити можливості аналізу семантики складових сервісів та доповнити розглянуті вище підходи елементами інтелектуалізації описів сервісів на основі онтологій [8].

References

1. A. L. Lemos, F. Daniel, B. Benatallah, Web service composition: a survey of techniques and tools. *ACM Computing Surveys (CSUR)*, 2015, 48(3), 1-41.
2. B. Jang, M. Kim, G. Harerimana, J. W. Kim, Q-learning algorithms: A comprehensive classification and applications. *IEEE access*, 2019 7, 133653-133667.
3. M. Karakus, A. Duresi, Quality of service (QoS) in software defined networking (SDN): A survey. *Journal of Network and Computer Applications*, 2017, 80, 200-218.
4. V. Uc-Cetina, F. Moo-Mena, R. Hernandez-Ucan, Composition of web services using Markov decision processes and dynamic programming. *The Scientific World Journal*, 2015, (1), 545308.
5. S. Kalasapur, M. Kumar, B. A. Shirazi, Dynamic service composition in pervasive computing. *IEEE Transactions on parallel and distributed systems*, 2007, 18(7), 907-918.
6. A. Alla, M. Falcone, D. Kalise, An efficient policy iteration algorithm for dynamic programming equations. *SIAM Journal on Scientific Computing*, 2015, 37(1), A181-A200.
7. D. Zhao, H. Wang, K. Shao, Y. Zhu, (, December). Deep reinforcement learning with experience replay based on SARSA. In: 2016 IEEE symposium series on computational intelligence (SSCI) 2016, pp. 1-6.
8. J.V. Rogushina, A.Y. Gladun, V.V. Osadchiy, S.M. Pryima Ontological analysis into the Web Melitopol: Bogdan Hmelnytsky MDPU, 2015, 407 p.[in Ukrainian],

Одержано: 25.10.2024

Внутрішня рецензія отримана: 02.11.2024

Зовнішня рецензія отримана: 05.11.2024

Про авторів:

Гришанова Ірина Юріївна,
науковий співробітник,
ORCID
<http://orcid.org/0000-0003-4999-6294>.

Рогущина Юлія Віталіївна,
к.ф.-м.н., доцент,
старший науковий співробітник,
ORCID
<http://orcid.org/0000-0001-7958-2557>.

Місце роботи авторів:

Інститут програмних систем
НАН України,
тел. (+38) 066 5501999,
e-mail:
ladamandraka2010@gmail.com

Ю.М. Бердник, А.О. Скотаренко

ДО ПРОБЛЕМИ РОЗПІЗНАВАННЯ ОБ'ЄКТІВ НА ПРИСТРОЯХ З ОБМЕЖЕНИМИ РЕСУРСАМИ

Розглянуто проблему реалізації розпізнавання об'єктів на мікроконтролерах з обмеженими ресурсами. Зокрема, виклики, пов'язані з їхньою обчислювальною потужністю, обсягом пам'яті та енергоспоживанням. Запропоновано використання компактної архітектури нейронної мережі MobileNet, яка базується на алгоритмі зі знизеним обчислювальним навантаженням і забезпечує достатню продуктивність у середовищах із обмеженими ресурсами. Такий підхід дає змогу виконувати задачі класифікації зображень на недорогих мікроконтролерах. Хоча технології, які розглядаються, є відомими самостійно, їхнє комплексне застосування для конкретних задач класифікації на мікроконтролерах, таких як розпізнавання об'єктів, залишається мало дослідженим. У статті наведено детальний опис усіх етапів розробки: підготовки даних, налаштування параметрів моделі, а також застосування спеціалізованих методів масштабування зображень для зменшення обчислювального навантаження. Практичну частину статті присвячено вирішенню прикладної задачі розроблення системи розпізнавання полуниць із визначенням ступеня їхньої стиглості на основі мікроконтролера ESP32. Результати дослідження демонструють ефективність підходу для різних прикладних задач і підтверджують можливість інтеграції технологій комп'ютерного зору у пристрої з обмеженими ресурсами. Загалом робота доводить, що сучасні технології машинного навчання стають доступними навіть для найменш потужних апаратних платформ, розширюючи їхні можливості та сфери використання. Ключові слова: розпізнавання об'єктів, класифікація, нейронна мережа, архітектура MobileNet, мікроконтролер.

У.М. Berdnyk, A.O. Skotarenko

ON THE PROBLEM OF OBJECT RECOGNITION ON DEVICES WITH LIMITED RESOURCES

The problem of implementing object recognition on microcontrollers with limited resources, particularly the challenges related to their computational power, memory capacity, and energy consumption is addressed. The use of the compact MobileNet neural network architecture is proposed, which is based on an algorithm with reduced computational load and ensures sufficient performance in resource-constrained environments. This approach enables image classification tasks to be performed on low-cost microcontrollers. Although the technologies discussed are well-known individually, their comprehensive application for specific classification tasks on microcontrollers, such as object recognition, remains underexplored. The article provides a detailed description of all development stages, including data preparation, model parameter tuning, and the use of specialized image scaling methods to reduce computational load. The practical part of the article is devoted to developing a strawberry recognition system to determine their ripeness level using the ESP32 microcontroller. The research results demonstrate the effectiveness of the approach for different applied tasks and confirm the feasibility of integrating computer vision technologies into resource-limited devices. Overall, the work proves that modern machine learning technologies are becoming accessible even to the least powerful hardware platforms, expanding their capabilities and areas of application.

Key words: object recognition, classification, neural network, MobileNet architecture, microcontroller.

Вступ

У сучасних реаліях нейронні мережі відіграють ключову роль у розвитку технологій. Розпізнавання об'єктів є важливою функцією у сучасних IoT-застосунках, автономних системах і робототехніці. Успішна інтеграція цієї технології дозволяє автоматизувати виявлення, класифікацію та відстеження об'єктів у реальному часі. Од-

нак інтеграція моделей нейронних мереж на пристроях із обмеженим обсягом пам'яті та обчислювальними ресурсами може стикатися з певними труднощами.

Існує два основні підходи до розпізнавання об'єктів залежно від використовуваних типів датчиків: системи з візуалізацією навколишнього середовища (каме-

ри/відео) та системи без візуалізації навколишнього середовища (ПЧ-датчики, лазерні, ультразвукові, Bluetooth або GPS) [1]. У цій статті розглянуто візуальне розпізнавання об'єктів на мікроконтролерах з обмеженими ресурсами.

Дана стаття пропонує підхід до вирішення проблеми реалізації розпізнавання об'єктів на пристроях із обмеженими ресурсами за допомогою технології MobileNet — компактної архітектури нейронної мережі, оптимізованої для роботи на пристроях із низькою продуктивністю. Для спрощення розробки використовується платформа Edge Impulse, яка дозволяє автоматизувати тренування, оптимізацію та розгортання моделей на мікроконтролерах. Обраний підхід демонструє, як можна подолати основні обмеження апаратного забезпечення, зберігши прийнятну точність і швидкість обробки.

Попри те, що кожна з цих технологій є самостійно відомою, їхнє поєднання для вирішення конкретної задачі класифікації зображень на пристроях із обмеженим апаратним потенціалом, таких як IoT-системи, є рідкісним у літературі.

Стаття може бути корисною для інженерів і розробників, оскільки включає детальний опис процесу підготовки даних, налаштування параметрів моделей і використання конкретних методів масштабування зображень.

Описаний підхід також підкріплюється прикладом вирішення прикладної задачі. Такий приклад не лише ілюструє ефективність підходу, а й демонструє його прикладну цінність для вузькоспеціалізованих завдань.

Основна мета — продемонструвати, що навіть у середовищах з обмеженими ресурсами можна реалізувати ефективні рішення для розпізнавання об'єктів.

1. Вибір способу реалізації розпізнавання об'єктів на мікроконтролері

Обмежені ресурси мікроконтролерів створюють багато технічних викликів, які змушують розробників шукати компроміс між продуктивністю та точністю. Насампе-

ред мікроконтролери, які мають дуже обмежений обсяг оперативної та постійної пам'яті. Типовий обсяг оперативної пам'яті у найпродуктивніших чіпах становить лише кілька мегабайтів [2], тоді як сучасні моделі для розпізнавання об'єктів, такі як MobileNet або YOLO, вимагають десятків чи навіть сотень мегабайтів для зберігання ваги та виконання обчислень. Через це навіть компактні моделі важко завантажити чи виконувати на таких пристроях, не кажучи вже про обробку зображень у реальному часі. Додатково низька тактова частота мікроконтролерів (зазвичай до 240 МГц) та відсутність спеціалізованих обчислювальних блоків, таких як GPU або TPU, призводять до тривалого часу виконання завдань навіть для невеликих моделей [2]. Це особливо критично для завдань у реальному часі, де навіть затримка в кілька секунд може бути непринятною.

Ще одним обмеженням є низьке енергоспоживання, яке є ключовим для автономних пристроїв. Розпізнавання об'єктів є завданням з високою обчислювальною складністю, що може значно зменшити тривалість роботи від батареї. Використання навіть оптимізованих моделей може швидко розрядити пристрій, особливо якщо задачі виконуються постійно. Крім того, обмежена пропускна здатність введення/виведення, характерна для мікроконтролерів, уповільнює передачу даних із сенсорів, таких як камери, до процесора, додаючи затримки до загального процесу розпізнавання.

Існує кілька підходів до вирішення цього завдання, кожен з яких має свої переваги і недоліки.

Одним із популярних підходів є серверна обробка, за якої мікроконтролер використовується лише для збору даних, таких як зображення, а всі обчислення виконуються на потужному сервері або в хмарному середовищі. Це дозволяє застосовувати складні моделі глибокого навчання, які забезпечують високу точність і можливість обробки великих обсягів даних. Однак серверна обробка має серйозні обмеження. По-перше, вона вимагає стабільного підключення до мережі, що не завжди можливо в автономних системах, як-от, у відда-

лених регіонах або в пристроях, які працюють у польових умовах. По-друге, передача даних на сервер і назад може викликати значні затримки, що робить цей підхід непридатним для завдань реального часу, таких як виявлення небезпечних ситуацій у системах безпеки чи управління робототехнікою [3]. Нарешті підтримка серверної інфраструктури чи оплата хмарних послуг може значно збільшити витрати, що не завжди виправдано у випадку компактних і недорогих пристроїв.

Іншим підходом є використання додаткових апаратних рішень, таких як спеціалізовані модулі для прискорення обчислень. Ці пристрої дозволяють виконувати складні нейронні мережі на периферійних пристроях, забезпечуючи високу швидкість і точність обробки даних. Вони особливо ефективні для роботи у реальному часі та автономних системах. Проте цей підхід також має недоліки. Використання додаткового обладнання збільшує загальну вартість системи, що може бути неприйнятним для недорогих пристроїв або масового виробництва. Крім того, інтеграція таких рішень вимагає додаткових ресурсів для налаштування, розробки та тестування. Апаратні модулі також збільшують струм спо-

живання та розміри пристрою, що може бути критичним для компактних пристроїв, таких як дрони чи переносні пристрої. Енергоспоживання цих рішень, хоча й оптимізоване, все одно часто перевищує можливості стандартних акумуляторів, примушуючи шукати компроміс між розмірами батареї і часом автономної роботи.

Третій підхід — використання оптимізованих моделей глибокого навчання, таких як MobileNet, Tiny-YOLO чи SqueezeNet. Ці архітектури спеціально розроблені для пристроїв із невеликими ресурсами для виявлення об'єктів з хорошою продуктивністю в реальному часі та високою точністю [4, 5, 6]. Основними перевагами цього підходу є можливість обробки даних безпосередньо на пристрої, що усуває залежність від підключення до мережі та наявності затримки.

MobileNet — це спрощена архітектура, яка використовує згортки, що розділяються по глибині, для побудови легких глибоких згорткових нейронних мереж і забезпечує ефективну модель для мобільних і вбудованих систем [7].

Структура MobileNet базується на фільтрах, що розділяються по глибині, як показано на рис. 1.

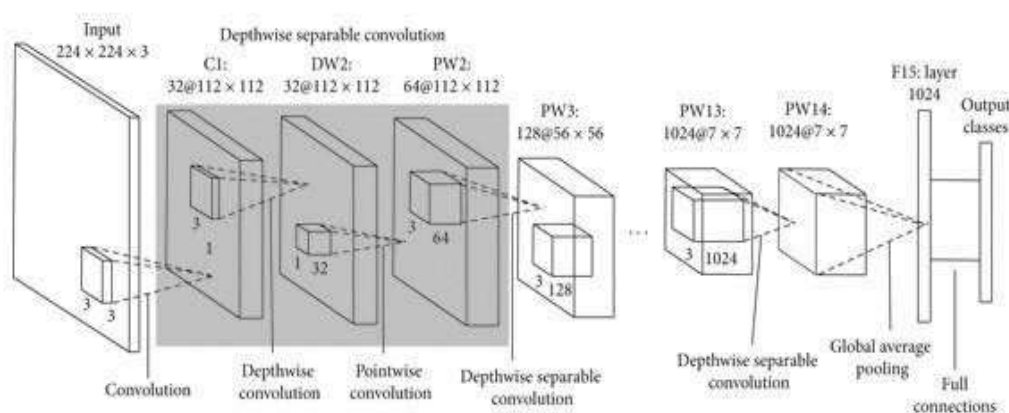


Рис. 1. Структура нейронної мережі MobileNet [8]

MobileNetV2 є значним удосконаленням у порівнянні з MobileNetV1 і впроваджує найсучасніші технології мобільного візуального розпізнавання, включаючи класифікацію, виявлення об'єктів і семантичну сегментацію. Архітектура MobileNetV2 заснована на перевернутій залишковій структурі, де вхід і вихід залишкового блоку є то-

нкими вузькими шарами, протилежними традиційним моделям залишків, які використовують розширені представлення у вхідних даних, а MobileNetV2 використовує легкі глибокі згортки для фільтрації функцій у проміжному шарі розширення. Загалом архітектура MobileNetV2 містить початковий повністю згорнутий рівень із 32 фі-

льтрами, за якими слідує 19 залишкових рівнів вузьких місць [9]. У роботі далі використовуватиметься зсув на 2 ($\text{stride}=2$), адже його основна ідея використання — це зменшення розмірів зображення (зменшення ширини та висоти), що допомагає зменшити обчислювальні витрати.

Проте навіть оптимізовані моделі вимагають спеціалізованих фреймворків і ретельного налаштування для ефективної роботи на мікроконтролерах.

Саме тому ще один підхід, який активно використовується в поєднанні з оптимізованими моделями, — це фреймворки для обробки на мікроконтролерах. Сучасні фреймворки, такі як TensorFlow Lite for Microcontrollers, CMSIS-NN і Edge Impulse, забезпечують інструменти для створення, оптимізації та розгортання моделей машинного навчання (ML) на пристроях із обмеженими ресурсами. Їхня мета — забезпечити ефективне виконання моделей машинного навчання на пристроях із малою пам'яттю та низькою обчислювальною потужністю.

Edge Impulse — це платформа повного циклу розробки, яка робить процес створення, розгортання та масштабування вбудованих програм з підтримкою машинного навчання простішим і швидшим, надаючи можливість розробникам створювати та оптимізувати рішення з реальними даними [10].

У статті запропоновано реалізацію третього і четвертого підходів — використання оптимізованих моделей глибокого навчання та фреймворку Edge Impulse. Такий вибір пояснюється низькою кількістю факторів. MobileNet забезпечує високу точність і компактність, що робить її придатною для мікроконтролерів із обмеженими ресурсами. Edge Impulse, у свою чергу, автоматизує складні етапи оптимізації та розгортання моделі, що значно прискорює процес розробки та мінімізує ризик помилок. Крім того, обробка даних на пристрої дозволяє забезпечити автономність, низькі затримки та підвищену безпеку, оскільки дані не потрібно передавати в мережу. Тож, запропоноване рішення поєднує в собі ефективність, простоту впровадження та відповідність вимо-

гам до компактних пристроїв із низьким енергоспоживанням.

2. Розгортання моделі розпізнавання об'єктів на мікроконтролері ESP32

Для одного з можливих рішень проблеми розпізнавання об'єктів за мікроконтролер було обрано обчислювальний модуль ESP32-CAM на основі мікроконтролера ESP32.

ESP32-CAM — це доступний і компактний модуль із можливостями Wi-Fi та Bluetooth, який також має вбудовану камеру [2]. Однак його апаратні обмеження ставлять значні виклики для виконання нейронних мереж. Основні обмеження можна виділити такі: незначний обсяг оперативної пам'яті, малий обсяг flash-пам'яті та невеликі обчислювальні потужності.

ESP32-CAM оснащений 520 КБ внутрішньої оперативної пам'яті та додатковими 8 МБ зовнішньої PSRAM [2]. Хоча PSRAM може використовуватися для зберігання більших обсягів даних, вона повільніша за внутрішню RAM і не завжди підходить для обчислювально-інтенсивних завдань. Цей обсяг пам'яті є дуже обмеженим для завантаження та виконання більшості моделей нейронних мереж, які часто потребують десятків або сотень мегабайтів для обробки великих масивів даних.

За замовчуванням ESP32-CAM має 32 Мбіт (4 МБ) флеш-пам'яті, де зберігаються машинні коди програми [2]. Це створює обмеження для завантаження моделей нейронних мереж, особливо великих. Флеш-пам'ять не придатна для виконання динамічних обчислень, тому обмеження її обсягу є критичним, якщо потрібно зберігати модель нейронної мережі на самому пристрої.

ESP32-CAM використовує процесор із двома ARM-ядрами і тактовою частотою до 240 МГц, який не призначений для інтенсивних обчислень, таких як виконання великих нейронних мереж [2].

Відсутність спеціалізованих обчислювальних блоків, таких як GPU чи TPU, обмежує можливість виконання обчислень із плаваючою точкою, які часто використовуються в нейронних мережах.

Тим не менше, такий модуль дуже зручний для простого розпізнавання об'єктів. Він має вбудовану камеру OV2640 та підтримує стиснений формат вихідного зображення JPEG [2].

На платформі розробки Edge Impulse можна одразу обрати під яку апаратну платформу буде виконуватися розробка. Серед запропонованих варіантів є і популярний ESP32-CAM. Далі було обрано версію мережі MobileNet, а саме MobileNetV2 0.35 на базі алгоритму зі зниженим обчислювальним навантаженням FOMO. Edge Impulse FOMO (швидші об'єкти, більше об'єктів) — це новий алгоритм машинного навчання, який забезпечує виявлення об'єктів на пристроях із жорсткими обмеженнями, що дозволяє підраховувати кілька об'єктів і знаходити їхнє розташування на зображенні в режимі реального часу, використовуючи до 30 разів менше процесорної потужності та пам'яті, ніж MobileNet SSD або YOLOv5 [11].

Для прикладу було виконано розпізнавання ягоди полуниці (рис. 2) та її класифікація на стиглу та незрілу, що може бути використано для створення недорогих агророботів для збору полуниць. Для створення навчального набору даних було використано існуючі зображення ягоди полуниці з відкритих джерел. Загалом для створення наборів було використано 120 зображень, які були поділені на 90 і 30 (75%/25%) на тренувальні та тестові набори відповідно. Всі вони були вручну проіндексовані та промарковані відповідно до стиглості ягід на зображеннях.

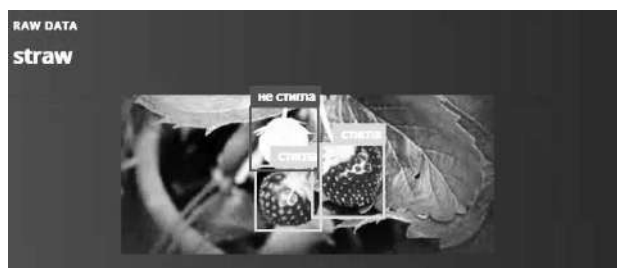


Рис. 2. Робота алгоритму класифікації полуниць

У процесі обробки зображень, вони були приведені до розміру 96 x 96. На основі отриманих даних було здійснено їх пе-

ретворення у набір спільних атрибутів. Після маркування наступним кроком стало створення та навчання моделі. На цьому етапі визначається тип даних для обробки, задаються параметри розміру зображень та метод їх масштабування. Водночас зберігається або змінюється співвідношення сторін залежно від потреб проекту. У Edge Impulse передбачено три способи досягнення необхідних розмірів: Squash, Fit shortest axis і Fit longest axis. Для розробки цієї моделі було обрано останній метод Fit longest axis, який додає чорні смуги (так звані "letterbox") вздовж коротшої осі зображення для досягнення потрібного співвідношення сторін. Потім зображення масштабується до визначеного розміру шляхом інтерполяції, забезпечуючи збереження пропорцій. На основі цих налаштувань модель виконує розпізнавання об'єктів і класифікує їх у дві категорії: «стигли» та «нестиглі» [12].

Результати створення характеристик наведені на рис. 3.

Як можна побачити з рис. 3, немає двох чітко виокремлених класів із тренувальних зображень. Причиною цього є наявність одночасно і стиглих, і не стиглих полуниць на одному зображенні.

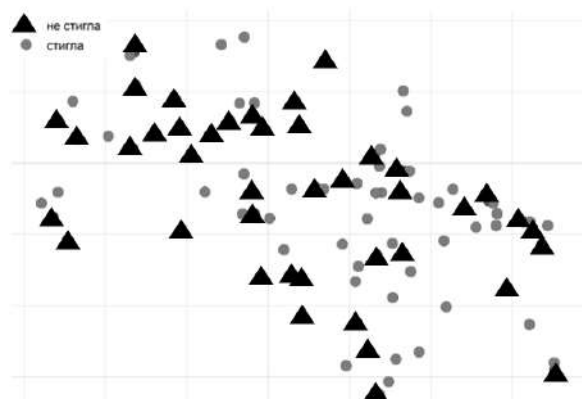


Рис. 3. Результати класифікації

Після налаштувань перед навчанням необхідно ще задати налаштування обробки нейронної мережі. Тут вказується кількість тренувальних циклів, чи використовувати навчений оптимізатор, навчальний крок (Learning rate), вид процесора для навчання, чи виконувати випадкове перетворення даних та інші додаткові умови. Для цієї системи були обрані такі параметри:

- number of training cycles: 70;
- use learned optimizer: ні (використовується стандартний оптимізатор);
- learning rate: 0.001;
- training processor: CPU (процес навчання буде здійснюватися на центральному процесорі);
- data augmentation: так (під час навчання будуть застосовуватися техніки аугментації даних, такі як обертання, зміна яскравості, масштабування тощо. Це допомагає покращити узагальнення моделі, зменшуючи перенавчання на тренувальному наборі даних) [12].

3. Результати роботи

На рис. 4 наявна інформація про загальну продуктивність моделі на валідаційній вибірці, зокрема, F1 Score дорівнює 74.8%. F1 Score — це середньогармонічне значення точності (precision) та повноти (recall), яке показує баланс між цими двома метриками. Значення 74.8% вказує на хорошу продуктивність моделі [13]. Далі зображена матриця плутанини (confusion matrix) на валідаційній вибірці. Вона показує, як добре модель класифікує зразки в кожну з категорій. Рядки представляють фактичні класи, а стовпці — передбачувані класи [12].

На ній видно 100% зразків, які фактично є «Фоном» і були правильно класифіковані як «Фон».

Для рядка «Нестигла» 40% зразків, які фактично є «Нестигла», були неправильно класифіковані як «Фон». Однак ці неправильні результати є не критичними для таких систем як, наприклад, робот для збору полуниць, адже знайшовши нестиглу полуницю, система не має нічого робити. 60% зразків, які фактично є «Нестигла», були правильно класифіковані як «Нестигла». І найголовніше для даної системи — 0% зразків, які фактично є «Нестигла», були неправильно класифіковані як «Стигла».

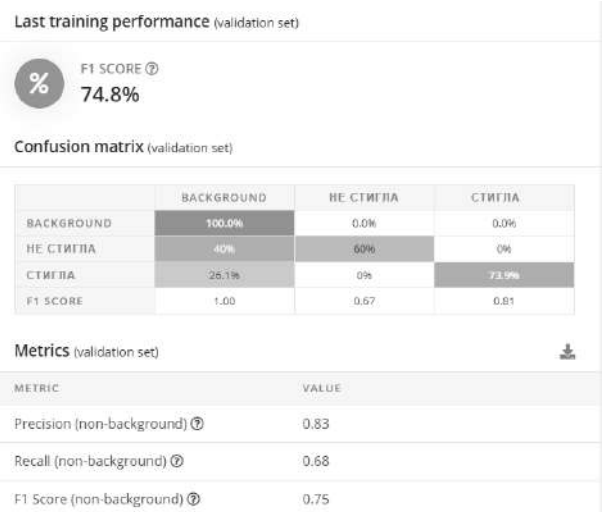


Рис. 4. Результати навчання моделі

Відповідні позитивні результати видно і в рядку «Стигла» у матриці плутанини: 26.1% зразків, які фактично є «Стигла», були неправильно класифіковані як «Фон»; 0% зразків, які фактично є «Стигла», були неправильно класифіковані як «Нестигла»; 73.9% зразків, які фактично є «Стигла», були правильно класифіковані як «Стигла». У цій же матриці вказані F1 Score для кожного класу: 1, 0.67, 0.81 до «Фон», «Стигла» та «Нестигла» відповідно.

Також результати оцінки містять такі метрики: Precision (non-background) або точність — відсоток правильних позитивних передбачень серед усіх позитивних передбачень [13]; Recall (non-background) або повнота — це відсоток правильних позитивних передбачень серед усіх фактичних позитивних зразків [13]; F1 Score (non-background) — це середньогармонічне значення точності та повноти для зразків, які не є фоном.

На рис. 5 наведені результати тестування моделі виявлення об'єктів. Там вказана загальна точність тестування 80%, точність Precision 94%, повнота Recall 87%, F1 Score 90%. Модель демонструє високу точність та повноту, що вказує на її здатність коректно виявляти більшість об'єктів і рідко видавати хибнопозитивні результати. У цілому, F1 оцінка 0.9 показує, що модель добре збалансована між точністю та повнотою. Проте загальна точність 80% вказує, що все ще є простір для поліпшення. Також зображена візуалізація ознак, де на графіку показано правильні і неправильні

передбачення моделі [12]. Приклад передбачення моделі наведено на рис. 6.

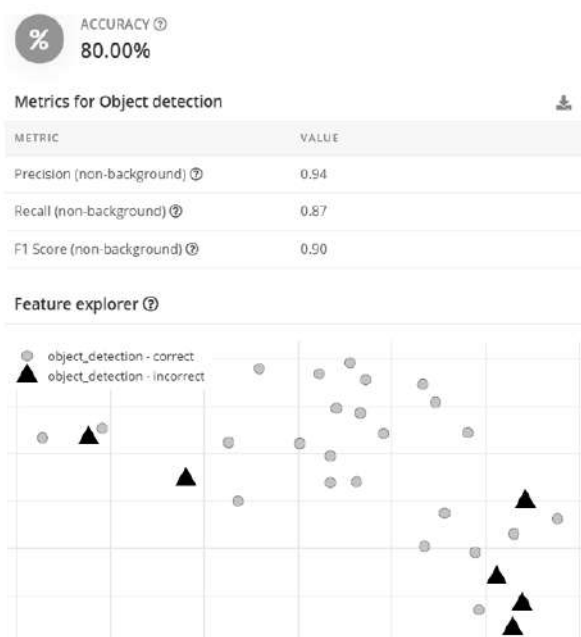


Рис. 5. Результати тестування моделі

Створена модель може розпізнавати полуницю в двох станах: стигла та нестигла. Наступним етапом став експорт моделі та конвертація її у вид, який можна завантажити на мікроконтролер. Для реалізації цього було конвертовано модель у бібліотеку та завантажено. Таким чином підключивши її, можна дописати потрібний код для подальшого керування периферійними пристроями та за потреби легко замінити модель на іншу [12].

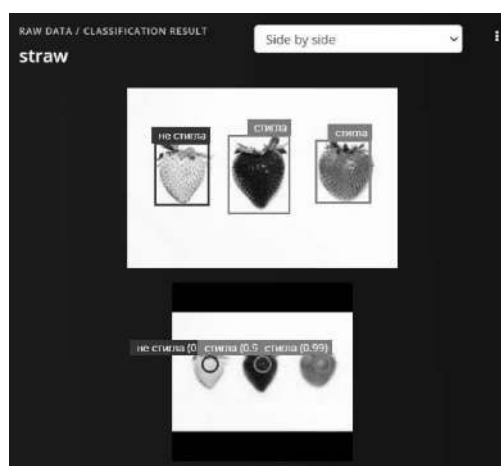


Рис. 6. Наочна демонстрація роботи моделі

Після імпортування моделі в середовище програмування плат Arduino IDE [14], було розроблено програму, що виводить

поточну інформацію про розпізнавання об'єктів та їх класів у консоль послідовного монітору (рис. 7).



Рис. 7. Вигляд роботи системи

Висновки

У статті розглянуто ключові виклики, що виникають під час реалізації розпізнавання об'єктів на мікроконтролерах з обмеженими ресурсами, а також запропоновано ефективне рішення, яке поєднує використання оптимізованої моделі MobileNet на основі алгоритму зі зниженим обчислювальним навантаженням і платформи Edge Impulse. Це дослідження продемонструвало, що навіть у середовищах із низькою обчислювальною потужністю можна створювати ефективні ML-рішення, які відповідають вимогам щодо точності, продуктивності та автономності.

Використання MobileNet на основі алгоритму зі зниженим обчислювальним навантаженням дозволило подолати проблему обмежених ресурсів завдяки її компактній архітектурі, що зменшує кількість обчислень за рахунок глибоких розділених згорток. Такий підхід забезпечив можливість запускати модель навіть на пристроях із мінімальними апаратними характеристиками, водночас зберігаючи прийнятний рівень точності. До того ж інтеграція з Edge Impulse значно спростила процес розробки. Ця платформа забезпечила інструменти для автоматизації таких складних етапів, як оптимізація моделі, її квантування навчання та інтеграція з мікроконтролером.

Завдяки цьому вдалося суттєво скоротити час розробки та уникнути багатьох потенційних помилок.

Ще однією важливою перевагою стала можливість локальної обробки даних безпосередньо на пристрої. Це рішення дозволило мінімізувати затримки, знизити залежність від зовнішніх мережевих ресурсів і забезпечити додаткову безпеку, оскільки дані не потребували передачі на сервери. Такий підхід особливо важливий для завдань реального часу, де затримка може суттєво вплинути на ефективність роботи системи.

Розроблена модель успішно виконала поставлену задачу — класифікацію об'єктів безпосередньо на мікроконтролері у дві категорії: «стигли» та «нестиглі» з точністю 80%. Це підтвердило ефективність запропонованого підходу, особливо для прикладних застосувань.

Дане дослідження відкриває нові перспективи для створення доступних і енергоефективних IoT-рішень, що здатні працювати автономно. Запропонований підхід може знайти широке застосування в таких галузях, як розумне сільське господарство, промислова автоматизація, системи моніторингу та переносні пристрої. Загалом робота доводить, що сучасні технології машинного навчання стають доступними навіть для найменш потужних апаратних платформ, розширюючи їхні можливості та сфери використання.

Подальші дослідження потрібні для визначення зв'язку між точністю розпізнавання, потрібним обсягом пам'яті та тактовою частотою, необхідною для забезпечення достатньої швидкості обробки зображень. Також потрібно дослідити способи реалізації запропонованого підходу на інших мікроконтролерах, зокрема, STM32.

Література

1. Ashiq, F.; Asif, M.; Bin Ahmad, M.; Zafar, S.; Masood, K.; Mahmood, T.; Mahmood, M.T.; Lee, I.H. *CNN-Based Object Recognition and Tracking System to Assist Visually Impaired People*: IEEE Access, 2022. 14819-14821 с.
2. ESP32-CAM: Technical Specification. URL: https://docs.sunfounder.com/projects/galaxy-rvr/en/latest/hardware/cpn_esp_32_cam.html (Дата звернення: 17.11.2024)
3. What are the pros and cons of AI (compared to traditional computing). URL: <https://www.ibm.com/think/insights/artificial-intelligence-advantages-disadvantages> (Дата звернення 17.11.2024)
4. Wei Fang, Lin Wang, Peiming Ren. *Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments*: IEEE Access, 2019. 1935-1938 с.
5. Yahui Nan, Jianguo Ju, Qingyi Hua, Haoming Zhang, Bo Wang: *A-MobileNet: An approach of facial expression recognition*: Alexandria Engineering Journal, 2021. 4435-4437 с.
6. SqueezeNet: *The Key to Unlocking the Potential of Edge Computing*. URL: <https://medium.com/sfu-csmp/squeezenet-the-key-to-unlocking-the-potential-of-edge-computing-c8b224d839ba> (Дата звернення 18.11.2024)
7. Howard A. G., Zhu M., Chen B. et al. *Mobilenets: efficient convolutional neural networks for mobile vision application*: Cornell University, 2017. 1-5с.
8. Wei Wang, Yutao Li, Ting Zou, Xin Wang, Jieyu You, Yanhong Luo: *A Novel Image Classification Approach via Dense-MobileNet Models*: Hindawi, 2020 3с.
9. MobileNetV2. URL: <https://paperswithcode.com/method/mobilenet-v2> (Дата звернення 18.11.2024)
10. EDGE IMPULSE. URL: <https://edgeimpulse.com/about> (Дата звернення 18.11.2024)
11. FOMO: Object detection for constrained devices. URL: <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/object-detection/fomo-object-detection-for-constrained-devices> (Дата звернення 18.11.2024)
12. Скотаренко А.О. Система автоматичного збору полуниць: бак.р. ...бакалавра: Київ, 2024. 40-52 с.
13. f1_score. URL: https://scikit-learn.org/1.5/modules/generated/sklearn.metrics.f1_score.html (Дата звернення 19.11.2024)
14. Arduino IDE. URL: <https://www.arduino.cc/en/software> (Дата звернення 18.11.2024)

Одержано: 25.11.2024

Внутрішня рецензія отримана: 30.11.2024

Зовнішня рецензія отримана: 30.11.2024

Про авторів:

Бердник Юрій Михайлович,
аспірант, асистент кафедри
інформаційних систем та технологій
<https://orcid.org/0000-0002-0008-4748>

Скотаренко Анастасія Олександрівна,
студентка магістр
<https://orcid.org/0009-0006-1429-5729>

Місце роботи авторів:

НТУ України «Київський політехнічний
інститут імені Ігоря Сікорського»
Тел.: +38-044-204-86-10.
E-mail: y.berdnyk@kpi.ua,
skotarenko.anastasiiia@iitl.kpi.ua

С.С. Шкільняк

ПЕРШОПОРЯДКОВІ ЛОГІКИ З ЧАСТКОВИМИ ПРЕДИКАТАМИ ДЛЯ ПЕРЕВІРКИ ОЗНАЧЕНОСТІ ЗМІННОЇ

В роботі досліджено нові класи програмно-орієнтованих логічних формалізмів – чисті першопорядкові логіки часткових квазіарних предикатів без обмеження монотонності. Їхньою характерною особливістю є наявність спеціальних 0-арних композицій – часткових предикатів, які визначають наявність у вхідних даних компоненти з відповідним предметним іменем (змінною), тобто з'ясовують означеність цього імені. Такі предикати-індикатори необхідні для елімінації кванторів у логіках немонотонних предикатів. Запропоновано два різновиди логік із частковими предикатами-індикаторами: L_1^{∂} (із традиційними реномінаціями) та $L_{1\perp}^{\partial}$ (із розширеними реномінаціями). Описано композиційні алгебри та мови цих логік, досліджено відношення логічного наслідку. Розглянуто умови гарантованої наявності цих відношень, наведено їхні основні властивості. Увагу акцентовано на властивостях, пов'язаних із предикатами-індикаторами та з елімінацією кванторів. Властивості відношень логічного наслідку є семантичною основою побудови для пропонованих логік відповідних числень секвенційного типу, що буде зроблено в наступних роботах.

Ключові слова: логіка, семантика, частковий предикат, логічний наслідок

Shkilniak S. S.

FIRST-ORDER LOGICS WITH PARTIAL PREDICATS FOR CHECKING VARIABLE DEFINEDNESS

We study semantic properties of new classes of program-oriented logics of partial quasiary predicates without monotonicity restriction. A feature of these logics is the use of special 0-ary parametric compositions – partial predicates which checks whether a subject name (variable) has a value in a given data. Such predicates-indicators are needed for quantifier elimination: from formulas of the form $\exists x\Phi$ we come to formulas of the form $R_z^x(\Phi)$. To perform such elimination in logics of non-monotonic predicates, the condition of definedness of a name z is needed, meaning a component with the name z is contained in the input data. We propose two types of pure first-order logics with partial predicates-indicators: L_1^{∂} and $L_{1\perp}^{\partial}$. Logics $L_{1\perp}^{\partial}$ use extended renominations, while L_1^{∂} use traditional renominations. In this paper we describe composition algebras and languages of these logics, and introduce and investigate logical consequence relations for formulas and sets of formulas of the language: irrefutability (IR), truth (T), falsity (F) and strong (TF) logical consequences. Conditions that guarantee the logical consequence relation are considered, and their main properties are specified. Special attention is paid to the properties related to predicates-indicators and quantifier elimination. Logical consequence relations' properties are the semantic basis for sequent calculi's construction. Basic properties of a given logical consequence relation induce respective sequent forms for the corresponding calculus; properties that guarantee the logical consequence relation induce closedness conditions for sequents in this calculus. Construction of such sequent calculi is planned in the future works.

Keywords: logic, semantics, partial predicate, logical consequence.

Вступ

Апарат математичної логіки успішно використовується для опису й моделювання різноманітних предметних областей, систем штучного інтелекту, інформаційних та програмних систем (див., напр., [1]). Водночас обмеження класичної логіки предикатів, роблять вельми актуальною задачу побудови нових, програмно-орієнтованих логічних формалізмів.

Метою пропонованої роботи є дослідження нових класів програмно-орієнтованих логік – чистих першопорядкових композиційно-номінативних логік часткових квазіарних предикатів (ЧКНЛ) зі знятою умовою монотонності (еквітонності). У побудові виведень в секвенційних численнях таких логік необхідно робити елімінацію кванторів. Це означає еквівалент-

ний перехід у секвенційних формах від формул вигляду $\exists x\Phi$ до їхніх прикладів – формул вигляду $R_z^x(\Phi)$. В логіках немонотонних предикатів це можливо за умови означеності предметного імені (змінної) z , тобто наявності у вхідних даних компоненти з іменем z . Тому для елімінації кванторів у логіках немонотонних предикатів потрібні спеціальні предикати-індикатори наявності у вхідних даних компоненти з відповідним предметним іменем. Можна виділити два їхніх різновиди: тотальні предикати-індикатори Ez , які визначають наявність чи відсутність компоненти з іменем z , та часткові, які визначають лише наявність такої компоненти.

Низку ЧКНЛ з тотальними предикатами-індикаторами Ez описано, зокрема, в роботах [2–5]. В даній роботі ми пропонуємо логіки з частковими предикатами-індикаторами $\downarrow z$.

Залежно від використання традиційних [2, 3] чи розширених [4, 5] реномінацій виділимо два різновиди ЧКНЛ з частковими предикатами-індикаторами: L_1^Q (із традиційними реномінаціями) та $L_{1\perp}^Q$ (із розширеними реномінаціями). Традиційні реномінації є окремим випадком розширених реномінацій, тому логіки L_1^Q є окремим випадком логік $L_{1\perp}^Q$.

Описані в [3] ЧКНЛ, збагачені предикатами слабкої рівності $=_{xy}$ та строгої рівності \equiv_{xy} , використовують предикати Ez . Водночас в ЧКНЛ зі слабкою рівністю окремим предикати-індикатори не потрібні, адже $\downarrow z$, та $=_{zz}$ – це один і той же предикат!

В даній роботі досліджено семантичні властивості $L_{1\perp}^Q$ та L_1^Q . Описано мови цих логік, досліджено відношення логічного наслідку. Розглянуто умови гарантованої наявності цих відношень, наведено їхні основні властивості. Особливу увагу приділено властивостям, пов'язаним з предикатами $\downarrow z$ та з елімінацією кванторів. Такі властивості є семантичною основою побудови для $L_{1\perp}^Q$ та L_1^Q відповідних числень секвенційного типу.

1. Композиційні предикатні алгебри логік $L_{1\perp}^Q$ та L_1^Q

Для полегшення читання наведемо необхідні для подальшого викладу визначення. Поняття, які тут не визначені, тлумачимо в сенсі робіт [2, 5].

V - A -квазіарний предикат – це часткова неоднозначна, взагалі кажучи, функція $Q: V^A \rightarrow \{T, F\}$, V^A – множина всіх V - A -іменних множин (V - A -ІМ), $\{T, F\}$ – множина істиннісних значень, V та A – множини предметних імен (змінних) та предметних значень (базових даних). V - A -ІМ формально визначають [2] як однозначну функцію $d: V \rightarrow A$. Подаємо V - A -ІМ у вигляді $[v_i \quad a_i]_{i \in I}$, де $v_i \in V$, $a_i \in A$, $v_i \neq v_j$ при $i \neq j$.

Для V - A -ІМ вводимо (див. [5]) операції $\|_{-Z}$, де $Z \subseteq V$, та (розширеної) реномінації $r_{x_1, \dots, x_n, \perp, \dots, \perp}^{v_1, \dots, v_n, u_1, \dots, u_m}$, де $v_i, x_i, u_j \in V$, а символ $\perp \notin V$ означає відсутність значення:

$$\begin{aligned} \|_{-Z}(d) &= \{v \quad a \in d \mid v \notin Z\}; \\ r_{x_1, \dots, x_n, \perp, \dots, \perp}^{v_1, \dots, v_n, u_1, \dots, u_m}(d) &= \\ &= [v_1 \mapsto d(x_1), \dots, v_n \mapsto d(x_n)] \cup d \|_{-v_1, \dots, v_n, u_1, \dots, u_m}. \end{aligned}$$

Порядок пар імен у цьому позначенні неістотний. Наприклад, $r_{x,z,\perp}^{v,y,u}$ та $r_{x,\perp,z}^{v,y,u}$ – це позначення однієї й тієї ж операції. Замість $r_{x_1, \dots, x_n, \perp, \dots, \perp}^{v_1, \dots, v_n, u_1, \dots, u_m}$ стисло пишемо $r_{\vec{x}, \perp}^{\vec{v}, \vec{u}}$.

Кожний R -предикат Q однозначно задається парою таких множин:

- область істинності $T(Q) = \{d \mid T \in Q[d]\}$;
- область хибності $F(Q) = \{d \mid F \in Q[d]\}$.

V - A -квазіарний R -предикат Q :

- частковий однозначний, або P -предикат, якщо $T(Q) \cap F(Q) = \emptyset$;
- тотальний, або T -предикат, якщо $T(Q) \cup F(Q) = V^A$;
- TS -предикат, якщо $T(Q) \cap F(Q) = \emptyset$ та $T(Q) \cup F(Q) = V^A$;
- неспростовний, якщо $F(Q) = \emptyset$;
- виконуваний, якщо $T(Q) \neq \emptyset$;
- тотожно істинний (позн. T), якщо $F(Q) = \emptyset$ та $T(Q) = V^A$;
- тотожно хибний (позн. F), якщо $T(Q) = \emptyset$ та $F(Q) = V^A$;
- тотально невизначений (позн. \perp), якщо $T(Q) = F(Q) = \emptyset$;
- тотально амбівалентний (позн. Υ), якщо $T(P) = F(P) = V^A$;

- монотонний: $d_1 \subseteq d_2 \Rightarrow Q[d_1] \subseteq Q[d_2]$;
- антитонний: $d_1 \subseteq d_2 \Rightarrow Q[d_1] \supseteq Q[d_2]$.

Для P -предикатів монотонність стає еквітонністю: Q еквітонний, якщо з умови $Q(d) \downarrow$ та $d \subseteq d'$ випливає $Q(d') \downarrow = Q(d)$.

Класи V - A -квазіарних R -, P -, T -, TS -предикатів позначатимемо PrR^{V-A} , PrP^{V-A} , PrT^{V-A} , $PrTS^{V-A}$. Усі TS -предикати, окрім константних T та F , немонотонні.

Монотонні R -предикати, антитонні R -предикати, еквітонні P -предикати, антитонні T -предикати названо RM -, RA -, PE -, TA -предикатами. Класи V - A -квазіарних RM -, RA -, PE -, TA -предикатів позначають $PrRM^{V-A}$, $PrRA^{V-A}$, $PrPE^{V-A}$, $PrTA^{V-A}$.

Ім'я $x \in V$ неістотне для R -предиката Q , якщо для всіх $d_1, d_2 \in V A$ маємо:

$$d_1 \parallel_{-x} d_2 \Rightarrow Q[d_1] = Q[d_2].$$

Опишемо базові композиції L_{\perp}^Q .

Пропозиційні композиції (логічні зв'язки) \neg і \vee та композицію квантифікації (квантор) $\exists x$ задамо так:

$$T(\neg P) = F(P); F(\neg P) = T(P);$$

$$T(P \vee Q) = T(P) \cup T(Q); F(P \vee Q) = F(P) \cap F(Q);$$

$$T(\exists x P) = \{d \mid d \parallel_{-x} \cup x \quad a \in T(P)\};$$

$$F(\exists x P) = \{d \mid d \parallel_{-x} \cup x \quad a \in F(P)\}.$$

Композицію реномінації $R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ задамо через відповідну операцію реномінації:

$$R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(Q)[d] = Q[r_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(d)].$$

Традиційні [2, 3] операція реномінації $r_{\bar{x}}^{\bar{v}}$ та композиція реномінації $R_{\bar{x}}^{\bar{v}}$ є окремими випадками $r_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$ та $R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}$.

Спеціальні 0-арні композиції – часткові предикати-індикатори $\downarrow z$ – задамо так:

$$T(\downarrow z) = \{d \mid d(z) \downarrow\}; F(\downarrow z) = \emptyset.$$

Предикати $\downarrow z$ еквітонні. Усі $x \in V$ такі, що $x \neq z$, неістотні для $\downarrow z$.

Отримуємо такі множини базових композицій для L_{\perp}^Q та L_{\perp}^Q :

$$- C_{\perp}^Q = \{\neg, \vee, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}, \exists x, \downarrow z\} \text{ для } L_{\perp}^Q;$$

$$- C_{\perp}^Q = \{\neg, \vee, R_{\bar{x}}^{\bar{v}}, \exists x, \downarrow z\} \text{ для } L_{\perp}^Q.$$

Нагадаємо, що для тотальних предикатів-індикаторів Ez маємо (див. [2, 5]): $T(Ez) = \{d \mid d(z) \downarrow\}$ та $F(Ez) = \{d \mid d(z) \uparrow\}$.

$$\text{Звідси } T(\downarrow z) = T(Ez).$$

Зауважимо, що в ЧКНЛ зі слабкою рівністю [3] предикати-індикатори вже є:

$$T(=_{zz}) = \{d \mid d(z) \downarrow\} = T(\downarrow z), F(=_{zz}) = \emptyset = F(\downarrow z).$$

Це означає: предикати $\downarrow z$ та $=_{zz}$ збігаються. Асиметрія предикатів $\downarrow x$ щодо істинності та хибності індукує таку асиметрію в логіках L_{\perp}^Q та L_{\perp}^Q . Це надає логікам L_{\perp}^Q та L_{\perp}^Q певних ознак девіантності.

Властивості пропозиційних композицій та кванторів, не пов'язані з реномінаціями, аналогічні властивостям класичних логічних зв'язок та кванторів [6]. Маємо [4, 5] такі базові властивості, пов'язані з композицією реномінації: Ren , R , R_{\perp} , $R_{\perp} \cup$, $R_{\perp} \uparrow_1$, $R_{\perp} \uparrow_2$, $R_{\perp} \neg$, $R_{\perp} \vee$, $R_{\perp} R$, $R_{\perp} \exists s$, $R_{\perp} \exists$. Для традиційної реномінації маємо (див. [2]) Ren , R , RI , RU , $R\neg$, $R\vee$, RR , $R\exists s$, $R\exists$.

Властивості предикатів $\downarrow z$, пов'язані з логічними зв'язками та кванторами:

$$- \neg \downarrow z \vee \downarrow z = \downarrow z \rightarrow \downarrow z = \downarrow z \leftrightarrow \downarrow z = \downarrow z;$$

$$- \neg \downarrow z \& \downarrow z = \neg \downarrow z;$$

$$- \exists x \downarrow x = \forall x \downarrow x = T,$$

$$- \exists x \neg \downarrow x = \forall x \neg \downarrow x = F;$$

$$- \exists y \downarrow x = \forall y \downarrow x = \downarrow x \text{ за умови } x \neq y;$$

$$- \exists y \neg \downarrow x = \forall y \neg \downarrow x = \neg \downarrow x \text{ за умови } x \neq y.$$

Властивості реномінації предикатів-індикаторів $\downarrow z$:

$$R_{\perp} \downarrow z R_{\bar{x}, \perp}^{\bar{v}, \bar{u}, z}(z) = \wedge;$$

$$R_{\perp} \downarrow z R_{\bar{x}, \perp, y}^{\bar{v}, \bar{u}, z}(\downarrow z) = \downarrow y;$$

$$R_{\perp} \downarrow z R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(\downarrow z) = \downarrow z \text{ за умови } z \notin \{\bar{v}, \bar{u}\}.$$

Для традиційної реномінації маємо:

$$R_{\perp} \downarrow z R_{\bar{x}, y}^{\bar{v}, z}(\downarrow z) = \downarrow y;$$

$$R_{\perp} \downarrow z R_{\bar{x}}^{\bar{v}}(\downarrow z) = \downarrow z \text{ за умови } z \notin \{\bar{v}\}.$$

Маємо $T(\downarrow z) = T(Ez)$, тому наведені в [4, 5]) властивості, на яких базується елімінація кванторів, можна переформулювати із використанням предикатів $\downarrow z$.

$$T(R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x}(P)) \cap T(\downarrow y) \subseteq T(R_{\bar{v}, \perp}^{\bar{u}, \bar{w}}(\exists x P)) \quad (\exists T_{\perp})$$

$$F(R_{\bar{v}, \perp}^{\bar{u}, \bar{w}}(\exists x P)) \cap T(\downarrow y) \subseteq F(R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x}(P)) \quad (\exists F_{\perp})$$

Ці співвідношення переформулюємо, записуючи $F(\neg \downarrow y)$ замість $T(\downarrow y)$:

$$T(R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x}(P)) \cap F(\neg \downarrow y) \subseteq T(R_{\bar{v}, \perp}^{\bar{u}, \bar{w}}(\exists x P)) \quad (\exists T_{\neg \perp})$$

$$F(R_{\bar{v}, \perp}^{\bar{u}, \bar{w}}(\exists x P)) \cap F(\neg \downarrow y) \subseteq F(R_{\bar{v}, \perp, y}^{\bar{u}, \bar{w}, x}(P)) \quad (\exists F_{\neg \perp})$$

Семантичними моделями певного класу ЧКНЛ є композиційні предикатні системи вигляду $(V A, Pr^{V-A}, C_B)$, де C_B – множина базових композицій, Pr^{V-A} – клас квазіарних предикатів. Кожна така система задає дві алгебри: алгебру (структуру) да-

них $({}^V A, PrR^{V-A})$ та композиційну алгебру предикатів (PrR^{V-A}, C_B) . Для $L_{\perp\perp}^Q$ і L_{\perp}^Q отримуємо композиційні системи R -предикатів $({}^V A, PrR^{V-A}, C_{\perp\perp}^Q)$ і $({}^V A, PrR^{V-A}, C_{\perp}^Q)$ та композиційні алгебри предикатів $A^{\perp\perp Q} = (PrR^{V-A}, C_{\perp\perp}^Q)$ і $A^{\perp Q} = (PrR^{V-A}, C_{\perp}^Q)$.

Необхідна умова того, що певний клас квазіарних предикатів утворює алгебру – його замкненість щодо операцій алгебри. 0-арні композиції – предикати $\downarrow z$ – однозначні й монотонні (еквітонні), а це означає незамкненість класів T -, TS -, RA -, TA -предикатів щодо композицій $C_{\perp\perp}^Q$ та C_{\perp}^Q . Водночас щодо композицій $C_{\perp\perp}^Q$ та C_{\perp}^Q замкнені класи P -предикатів, а також RM - та PE -предикатів. Тому можна виділити такі підалгебри алгебр $A^{\perp\perp Q}$ та $A^{\perp Q}$:

$$A^{\perp\perp QP} = (PrP^{V-A}, C_{\perp\perp}^Q), A^{\perp QP} = (PrP^{V-A}, C_{\perp}^Q);$$

$$A^{\perp\perp QM} = (PrRM^{V-A}, C_{\perp\perp}^Q), A^{\perp QM} = (PrRM^{V-A}, C_{\perp}^Q);$$

$$A^{\perp\perp QPE} = (PrPE^{V-A}, C_{\perp\perp}^Q), A^{\perp QPE} = (PrPE^{V-A}, C_{\perp}^Q).$$

Логіки монотонних та еквітонних предикатів тут не розглядаємо, тому зосередимо увагу на алгебрах R -предикатів $A^{\perp\perp Q}$ і $A^{\perp Q}$ та P -предикатів $A^{\perp\perp QP}$ і $A^{\perp QP}$.

2. Мови логік $L_{\perp\perp}^Q$ та L_{\perp}^Q

Терми композиційної алгебри предикатів трактуємо як формули мови.

При зафіксованій множині базових композицій мови ЧКНЛ відрізняються множинами імен базових предикатів (сигнатурою) і способами запису формул.

Опишемо мову $L_{\perp\perp}^Q$. Використовуємо префіксну форму запису формул.

Алфавіт мови:

- множина V предметних імен (змінних),
- множина Ps предикатних символів,
- множина $Cs = \{\neg, \vee, R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}, \exists x, \downarrow z\}$ символів базових композицій.

Дамо індуктивне визначення множини Fr формул мови:

- Fa) $Ps \subseteq Fr$;
 - F \downarrow) $\{\downarrow z \mid z \in V\} \subseteq Fr$;
 - Fp) $\Phi, \Psi \in Fr \Rightarrow \neg\Phi \in Fr, \vee\Phi\Psi \in Fr$;
 - FR \perp) $\Phi \in Fr \Rightarrow R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}\Phi \in Fr$;
 - F \exists) $\Phi \in Fr \Rightarrow \exists x\Phi \in Fr$.
- Формули $p \in Ps$ та $\downarrow z$ – атомарні.

В мовах ЧКНЛ задаємо множину $V_T \subseteq V$ імен, неістотних для всіх $p \in Ps$ – тотально неістотних [2] імен.

Для визначення множин *гарантовано неістотних* для формул мови $L_{\perp\perp}^Q$ імен задамо функцію $v: Fr \rightarrow 2^V$. Для кожного $x \in V$ задаємо $v(\downarrow x) = V \setminus \{x\}$, для кожного $p \in Ps$ задаємо $v(p)$, водночас має виконуватись $V_T \subseteq v(p)$, а далі визначаємо так, як описано в [2].

Розширена сигнатура мови – це $\Sigma = (V, V_T, Cs, Ps)$.

$\Phi \in Fr$ – *CF-формула* (constant free), якщо Φ не містить символів композицій-констант, тобто не містить символів $\downarrow x$.

Для $\Gamma \subseteq Fr$ вводимо позначення:

- $\sigma(\Gamma)$ – це множина всіх $p \in Ps$, які входять до складу формул $\Phi \in \Gamma$;
- $nm(\Gamma)$ – це множина всіх $x \in V$, які фігурують у формулах $\Phi \in \Gamma$.
- $v(\Gamma) = \bigcup_{\Phi \in \Gamma} v(\Phi)$.

Зокрема, для $\Phi \in Fr$ вводимо позначення $\sigma(\Phi)$ та $nm(\Phi)$.

Мову $L_{\perp\perp}^Q$ інтерпретуємо на композиційних системах $CS = ({}^V A, PrR^{V-A}, C_{\perp\perp}^Q)$.

Символи базових композицій інтерпретуємо як відповідні композиції, символи $\downarrow x$ – як відповідні предикати-індикатори. Для позначення символами Ps базових предикатів задамо тотальне однозначне $I: Ps \rightarrow PrR^{V-A}$ та продовжимо його до відображення інтерпретації $I: Fr \rightarrow PrR^{V-A}$:

- I \neg) $I(\neg\Phi) = \neg(I(\Phi))$;
- I \vee) $I(\vee\Phi\Psi) = \vee(I(\Phi), I(\Psi))$;
- IR \perp) $I(R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}\Phi) = R_{\bar{x}, \perp}^{\bar{v}, \bar{u}}(I(\Phi))$;
- I \exists) $I(\exists x\Phi) = \exists x(I(\Phi))$.

Трійку $J = (CS, \Sigma, I)$ назвемо *інтерпретацією* мови $L_{\perp\perp}^Q$. Кожна (CS, Σ, I) визначається алгеброю із доданою сигнатурою $A = ({}^V A, PrR^{V-A}, \Sigma, I)$. Такі алгебри пов'язують мову логіки з алгеброю даних, вони визначають композиційні системи $CS = ({}^V A, PrR^{V-A}, C_{\perp\perp}^Q)$. Тому вважатимемо зазначені алгебри з доданою сигнатурою *інтерпретаціями* мови $L_{\perp\perp}^Q$ та скорочено позначатимемо їх $A = (A, I)$.

Мова L_{\perp}^Q визначається аналогічно мові L_{\perp}^Q , лише замість символів $R_{x,\perp}^{\bar{v},\bar{u}}$ використовуємо символи $R_x^{\bar{v}}$.

Предикат $J(\Phi)$ – значення формули Φ при інтерпретації J – позначимо Φ_J .

Ім'я $x \in V$ неістотне для формули Φ , якщо x неістотне для Φ_J для кожної J .

Теорема 1. Якщо $x \in v(\Phi)$, то x неістотне для формули Φ .

Для $\Phi \in Fr$ та $\Gamma \subseteq Fr$ визначимо $fu(\Phi) = V_T \setminus nm(\Phi)$ та $fu(\Gamma) = V_T \setminus nm(\Gamma)$.

Теорема 2. $y \in fu(\Phi) \Rightarrow y \in v(\Phi)$.

Наслідок. Нехай $\Gamma \subseteq Fr$ та $y \in fu(\Gamma)$, тоді y неістотне для кожної $\Phi \in \Gamma$.

В алгебрах R -предикатів $A^{\perp\perp Q}$ та A^{1Q} виділено підалгебри P -предикатів $A^{\perp\perp QP}$ та A^{1QP} , а також підалгебри RM -предикатів $A^{\perp\perp QM}$, A^{1QM} , та PE -предикатів $A^{\perp\perp QPE}$, A^{1QPE} . Це виділяє класи R -інтерпретацій та підкласи P -інтерпретацій, а також RM - та PE -інтерпретацій. Такі класи інтерпретацій називають *семантиками*.

Таким чином, в L_{\perp}^Q маємо семантики $\perp R^1$, $\perp P^1$, $\perp RM^1$, $\perp PE^1$. В L_{\perp}^Q маємо семантики R^1 , P^1 , RM^1 , PE^1 .

$\perp R^1$ та R^1 назвемо R -семантиками.

$\perp P^1$ та P^1 назвемо P -семантиками.

В цій роботі вивчаємо логіки без обмеження монотонності предикатів, тому семантики типів RM та PE не розглядаємо, увагу зосередимо на R - та P -семантиках.

Нехай α – певна семантика.

Формула Φ *неспростовна* за інтерпретації J , або J -неспростовна (позн. $J \models \Phi$), якщо предикат Φ_J – неспростовний. Формула Φ *неспростовна* в семантиці α (позн. $\alpha \models \Phi$), якщо $J \models \Phi$ при кожній $J \in \alpha$.

В логіках L_{\perp}^Q та L_{\perp}^Q неспростовними є усі предикати-індикатори $\downarrow x$.

Формула Φ *виконувана* за інтерпретації J , або J -виконувана, якщо предикат Φ_J виконуваний. Φ *виконувана* в семантиці α , якщо Φ виконувана за певної $J \in \alpha$.

Для R -семантик поняття виконуваної формули малозмістовне. Справді, в R -семантиці α візьмемо $J \in \alpha$ таку, що $p_J = Y$ для кожної $p \in Ps$, тому $T(\Phi_J) \neq \emptyset$ для кож-

ної $\Phi \in Fr$. Звідси кожна формула виконувана в R -семантиці α .

Формула Φ *тотожно істинна* при інтерпретації J (позн. $J \models_{id} \Phi$), якщо предикат $\Phi_J = T$. Формула Φ *тотожно істинна* в семантиці α (позн. $\alpha \models_{id} \Phi$), якщо $J \models_{id} \Phi$ при кожній $J \in \alpha$.

Подібним чином можна дати визначення тотожно хибної за інтерпретації J та тотожно хибної в семантиці α формули.

Якщо семантика α зафіксована, то замість $\alpha \models$, $\alpha \models_{id}$ пишемо \models , \models_{id} .

Маємо $J \models_{id} \Phi \Rightarrow J \models \Phi$, $\models_{id} \Phi \Rightarrow \models \Phi$.

Формули, які завжди інтерпретуються як константні предикати, назвемо *константними* формулами. Такими є:

- T -формули, які завжди інтерпретуються як T ; позначаємо їх $T\Phi$;
- F -формули, які завжди інтерпретуються як F ; позначаємо їх $F\Phi$;
- \perp -формули, які завжди інтерпретуються як \perp ; позначаємо їх $\perp\Phi$.

Приклад 1. 1) $\exists x \downarrow x$ є T -формулами;

2) $\exists x \neg \downarrow x$ є F -формулами;

3) $R_{x,\perp}^{\bar{v},\bar{u},z}(\downarrow z)$ та $R_{\perp}^z(\downarrow z)$ є \perp -формулами.

Остання властивість засвідчує істотну відмінність L_{\perp}^Q від L_{\perp}^Q .

Якщо ϑ – T -формула, то $\alpha \models_{id} \vartheta$.

Формули, які завжди інтерпретуються як частково константні предикати, назвемо *частково константними*. Такими є:

- rF -формули, в яких $T(\Phi_J) = \emptyset$ для всіх J ; зокрема, це F -формули.
- rT -формули, в яких $F(\Phi_J) = \emptyset$ для всіх J ; зокрема, це T -формули.

Для rT -формул Φ маємо $\alpha \models \Phi$.

До rT -формул в L_{\perp}^Q та L_{\perp}^Q належать символи $\downarrow z$. Справді, $F(\downarrow z_J) = \emptyset$ для всіх J .

3. Відношення логічного наслідку

На основі різних співвідношень між областями істинності та хибності предикатів можна ввести низку відношень логічного наслідку на множині формул мови. Спочатку задаємо (див. [2, 5]) такі “природні” відношення наслідку для двох формул Φ та Ψ за умови фіксованої інтерпретації J :

- істинний, або T -наслідок $J|=T$:
- $\Phi J|=T \Psi \Leftrightarrow T(\Phi_J) \subseteq T(\Psi_J)$;
- хибний, або F -наслідок $J|=F$:
- $\Phi J|=F \Psi \Leftrightarrow F(\Psi_J) \subseteq F(\Phi_J)$;
- сильний, або TF -наслідок $J|=TF$:
- $\Phi J|=TF \Psi \Leftrightarrow \Phi J|=T \Psi$ та $\Phi J|=F \Psi$;
- неспростовний, або IR -наслідок
- $J|=IR$: $\Phi J|=IR \Psi \Leftrightarrow T(\Phi_J) \cap F(\Psi_J) = \emptyset$.
- дуальний до IR , або DI -наслідок
- $J|=DI$: $\Phi J|=DI \Psi \Leftrightarrow F(\Phi_J) \cup T(\Psi_J) = V_A$.

Зазначені відношення індукують відповідні відношення еквівалентності формул Φ та Ψ за умови інтерпретації J . Задаємо їх за такою схемою:

$\Phi \sim^* \Psi$, якщо $\Phi J|=^* \Psi$ та $\Psi J|=^* \Phi$.

Для відношення $J \sim_{TF}$ маємо:

$\Phi \sim_{TF} \Psi \Leftrightarrow T(\Phi_J) = T(\Psi_J)$ та $F(\Phi_J) = F(\Psi_J)$.

Отже, $\Phi \sim_{TF} \Psi$ означає, що Φ_J та Ψ_J – це один і той же предикат.

Відношення логічного τ -наслідку в семантиці α та логічної еквівалентності в семантиці α задаємо за схемами:

$\Phi \alpha|= \tau \Psi$, якщо $\Phi J|= \tau \Psi$ для кожної $J \in \alpha$;

$\Phi \alpha \sim^* \Psi$, якщо $\Phi \sim^* \Psi$ для кожної $J \in \alpha$.

На основі відношень наслідку $J|=IR$, $J|=DI$, $J|=T$, $J|=F$, $J|=TF$ для двох формул при фіксованій J для семантик R^1 , P^1 (мова L_1^Q) та $\perp R^1$, $\perp P^1$ (мова $L_{1\perp}^Q$) загалом отримуємо такі відношення логічного наслідку:

- $R_{\downarrow}|=IR, R_{\downarrow}|=DI, R_{\downarrow}|=T, R_{\downarrow}|=F, R_{\downarrow}|=TF$;
- $P_{\downarrow}|=IR, P_{\downarrow}|=DI, P_{\downarrow}|=T, P_{\downarrow}|=F, P_{\downarrow}|=TF$;
- $R_{\perp\downarrow}|=IR, R_{\perp\downarrow}|=DI, R_{\perp\downarrow}|=T, R_{\perp\downarrow}|=F, R_{\perp\downarrow}|=TF$;
- $P_{\perp\downarrow}|=IR, P_{\perp\downarrow}|=DI, P_{\perp\downarrow}|=T, P_{\perp\downarrow}|=F, P_{\perp\downarrow}|=TF$.

Деякі з цих відношень вироджені.

Твердження 1. 1) Не існує CF -формул Φ та Ψ : $\Phi R_{\downarrow}|=IR \Psi$ чи $\Phi R_{\perp\downarrow}|=IR \Psi$.

2) Не існує CF -формул Φ та Ψ : $\Phi R_{\downarrow}|=DI \Psi$ чи $\Phi R_{\perp\downarrow}|=DI \Psi$ чи $\Phi P_{\downarrow}|=DI \Psi$ чи $\Phi P_{\perp\downarrow}|=DI \Psi$.

Візьмемо інтерпретацію J , в якій усі Ps інтерпретуються як Υ . Тоді $\Phi_J = \Upsilon$ та $\Psi_J = \Upsilon \Rightarrow T(\Phi_J) = T(\Psi_J) = F(\Phi_J) = F(\Psi_J) = V_A \Rightarrow T(\Phi_J) \cap F(\Phi_J) = V_A$. Це доводить п. 1.

Візьмемо інтерпретацію J , в якій усі Ps інтерпретуються о як λ . Тоді $\Phi_J = \lambda$ та $\Psi_J = \lambda \Rightarrow T(\Phi_J) = T(\Psi_J) = F(\Phi_J) = F(\Psi_J) = \emptyset \Rightarrow F(\Phi_J) \cup T(\Phi_J) = \emptyset$. Це доводить п. 2.

Таким чином, відношення типу DI та відношення $R_{\downarrow}|=IR, R_{\perp\downarrow}|=IR$ вироджені.

Розглянемо властивості, пов'язані з предикатами $\downarrow z$ та відношеннями $\alpha|=T$ і $\alpha|=F$ (тут α – одне з $R_{\downarrow}, R_{\perp\downarrow}, P_{\downarrow}, P_{\perp\downarrow}$).

Маємо $F(\downarrow x) = \emptyset$ та $\exists x \downarrow x = T$, звідки

Твердження 2. 1) для довільних $\Phi \in Fr$ та $y \in V$ маємо $\Phi \alpha|=F \downarrow y$ та $\Phi \alpha|=IR \downarrow y$, водночас $\exists x \downarrow x \alpha| \neq T \downarrow y$;

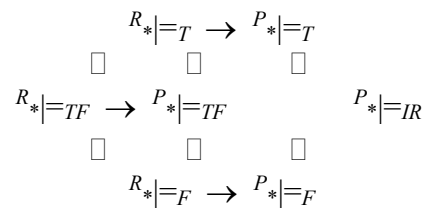
2) $\neg \downarrow y \alpha|=T \Phi$ та $\neg \downarrow y \alpha|=IR \Phi$ для довільних $\Phi \in Fr$ та $y \in V$, проте $\neg \downarrow y \alpha| \neq F \neg \exists x \downarrow x$;

3) $\downarrow x \alpha \sim_F \downarrow y$ та $\downarrow x \alpha \sim_{IR} \downarrow y$ для всіх $x, y \in V$.

Це засвідчує, що в L_1^Q та $L_{1\perp}^Q$ відношення типів T, F та TF істотно різні. Отже, в L_1^Q та $L_{1\perp}^Q$ маємо по 7 різних невироджених відношень логічного наслідку:

$P_{\downarrow}|=IR, P_{\downarrow}|=T, P_{\downarrow}|=F, P_{\downarrow}|=TF, R_{\downarrow}|=T, R_{\downarrow}|=F, R_{\downarrow}|=TF$;
 $P_{\perp\downarrow}|=IR, P_{\perp\downarrow}|=T, P_{\perp\downarrow}|=F, P_{\perp\downarrow}|=TF, R_{\perp\downarrow}|=T, R_{\perp\downarrow}|=F, R_{\perp\downarrow}|=TF$.

Графічно співвідношення між ними можна подати так (тут $*$ – це \downarrow чи $\perp\downarrow$, а замість \subseteq вживаємо \rightarrow):



Відношення логічного наслідку індукують відношення логічної еквівалентності. Отримуємо 7+7 відповідних відношень $P_* \sim_{IR}, P_* \sim_T, P_* \sim_F, P_* \sim_{TF}, R_* \sim_T, R_* \sim_F, R_* \sim_{TF}$.

Далі індекси \downarrow та $\perp\downarrow$ в позначенні відношень зазвичай опускаємо, із контексту буде зрозуміло, про яку семантику йде мова.

Теорема 3. Зазначені відношення логічного наслідку рефлексивні й транзитивні; відношення логічної еквівалентності рефлексивні, транзитивні та симетричні.

Описані в [2] властивості пропозиційного рівня справджуються для введених тут відношень в L_1^Q й $L_{1\perp}^Q$. Зокрема:

Теорема 4. 1) $\neg \Phi \& \Phi^P|=T \Psi$, проте $\neg \Phi \& \Phi^P| \neq F \Psi$;

2) $\Phi^P|=F \neg \Psi \vee \Psi$, проте $\Phi^P| \neq T \neg \Psi \vee \Psi$;

3) $\neg \Phi \& \Phi^P|=TF \neg \Psi \vee \Psi$;

4) $\neg \Phi \& \Phi^R| \neq TF \neg \Psi \vee \Psi$.

Основа еквівалентних перетворень формул – теорема еквівалентності. Вона виконується для відношень типів TF та IR , проте невірна для відношень типів T та F .

Теорема 5. Нехай Φ' отримано з формули Φ заміною деяких входжень Φ_1, \dots, Φ_n на Ψ_1, \dots, Ψ_n . Якщо $\Phi_1 \alpha \sim^* \Psi_1, \dots, \Phi_n \alpha \sim^* \Psi_n$, то $\Phi \alpha \sim^* \Phi'$.

Тут $\alpha \sim^*$ одне з $R \sim_{TF}, P \sim_{TF}, P \sim_{IR}$.

На основі властивостей предикатів $Ren, R, R_{\perp I}, R_{\perp U}, R_{\uparrow 1}, R_{\uparrow 2}, R_{\perp \neg}, R_{\perp \vee}, R_{\perp R}, R_{\perp \exists s}, R_{\perp \exists}$ (логіка $L_{\perp \perp}^Q$) та $Ren, R, RI, RU, R_{\perp \neg}, R_{\perp \vee}, RR, R_{\perp \exists s}, R_{\perp \exists}$ (логіка L_1^Q) отримуємо однойменні властивості формул. Для їхнього опису використаємо відношення типу \sim_{TF} .

Для формул мов L_1^Q та $L_{\perp \perp}^Q$ маємо специфічні властивості пов'язані з $\downarrow z$. Вони отримуються на основі наведених вище відповідних властивостей предикатів:

$$R_{\perp \downarrow \vee} R_{\bar{x}, \downarrow, y}^{\bar{v}, \bar{u}, z}(\downarrow z) \sim_{TF} \downarrow y;$$

$$R_{\perp \downarrow \wedge} R_{\bar{x}, \downarrow}^{\bar{v}, \bar{u}}(\downarrow z) \sim_{TF} \downarrow z \text{ за умови } z \notin \{\bar{v}, \bar{u}\};$$

$$R_{\downarrow \vee} R_{\bar{x}, y}^{\bar{v}, z}(\downarrow z) \sim_{TF} \downarrow y;$$

$$R_{\downarrow \wedge} R_{\bar{x}}^{\bar{v}}(\downarrow z) \sim_{TF} \downarrow z \text{ за умови } z \notin \{\bar{v}\}.$$

Пропозиційні властивості $\downarrow z$:

$$\alpha \models \downarrow z; \neg \downarrow z \vee \downarrow z \sim_{TF} \downarrow z; \neg \downarrow z \ \& \ \downarrow z \sim_{TF} \neg \downarrow z.$$

Властивості квантифікації:

$$- \alpha \models_{id} \exists x \downarrow x \text{ та } \alpha \models_{id} \forall x \downarrow x;$$

$$- \exists y \downarrow x \sim_{TF} \downarrow x \text{ та } \forall y \downarrow x \sim_{TF} \downarrow x \text{ за умови } x \neq y.$$

Відношення наслідку і логічного наслідку поширимо на пари множин формул. Назвемо їх відношеннями set-наслідку та логічного set-наслідку.

Нехай деяка $\Sigma \subseteq Fr$, нехай J – інтерпретація. Далі позначасмо:

$$T(\theta_J) \text{ як } T^\cap(\Sigma_J), \quad F(\theta_J) \text{ як } F^\cap(\Sigma_J),$$

$$T(\theta_J) \text{ як } T^\cup(\Sigma_J), \quad F(\theta_J) \text{ як } F^\cup(\Sigma_J).$$

Нехай $\Gamma, \Delta \subseteq Fr$.

$\Delta \in IR$ -наслідком Γ при інтерпретації J (позн. $\Gamma \models_{IR} \Delta$), якщо $T^\cap(\Gamma_J) \cap F^\cap(\Delta_J) = \emptyset$.

$\Delta \in DI$ -наслідком Γ при інтерпретації J (позн. $\Gamma \models_{DI} \Delta$), якщо $F^\cup(\Gamma_J) \cup T^\cup(\Delta_J) = \forall A$.

$\Delta \in T$ -наслідком Γ при інтерпретації J (позн. $\Gamma \models_T \Delta$), якщо $T^\cap(\Gamma_J) \subseteq T^\cup(\Delta_J)$.

$\Delta \in F$ -наслідком Γ при інтерпретації J (позн. $\Gamma \models_F \Delta$), якщо $F^\cap(\Delta_J) \subseteq F^\cup(\Gamma_J)$.

$\Delta \in TF$ -наслідком Γ при інтерпретації J (позн. $\Gamma \models_{TF} \Delta$), якщо $\Gamma \models_T \Delta$ та $\Gamma \models_F \Delta$.

Відношення логічного τ -set-наслідку в семантиці α задамо за схемою:

$\Gamma \alpha \models \tau \Delta$, якщо $\Gamma \models \tau \Delta$ для кожної $J \in \alpha$.

В R - та P -семантиках логік L_1^Q та $L_{\perp \perp}^Q$ маємо по 7 різних невироджених відношень логічного set-наслідку. Їхні назви та співвідношення між ними такі ж, як для відповідних відношень для двох формул.

Відношення типів DI та IR в R -семантиках вироджені. Відношення типів T, F та TF в R -семантиках істотно різні.

Твердження теореми 4 для відношень set-наслідку узагальнюється так.

Теорема 6. 1) $\Gamma, \neg \Phi \ \& \ \Phi \models_T \Psi, \Delta$, проте $\Gamma, \neg \Phi \ \& \ \Phi \not\models_F \Psi, \Delta$;

2) $\Gamma, \Phi \models_F \neg \Psi \vee \Psi, \Delta$, проте $\Gamma, \Phi \not\models_T \neg \Psi \vee \Psi, \Delta$;

3) $\Gamma, \neg \Phi \ \& \ \Phi \models_{TF} \neg \Psi \vee \Psi, \Delta$;

4) $\Gamma, \neg \Phi \ \& \ \Phi \not\models_{TF} \neg \Psi \vee \Psi, \Delta$.

Аналогом теореми еквівалентності для відношень є теорема заміни еквівалентних. Вона справджується для відношень логічної еквівалентності типів $R \sim_{TF}, P \sim_{TF}, P \sim_{IR}$ та відношень логічного наслідку відповідних типів: $R \models_{TF}, P \models_{\sim TF}, P \models_T, P \models_F, P \models_{IR}$ у випадку $R \sim_{TF}; P \models_{\sim TF}, P \models_T, P \models_F, P \models_{IR}$ у випадку $P \sim_{TF}; P \models_{IR}$ у випадку $P \sim_{IR}$.

Теорема 7. Нехай $\Phi \sim_\alpha \Psi$, тоді:

$$\Phi, \Gamma \models_\alpha \Delta \Leftrightarrow \Psi, \Gamma \models_\alpha \Delta;$$

$$\Gamma \models_\alpha \Delta, \Phi \Leftrightarrow \Gamma \models_\alpha \Delta, \Psi.$$

Для введених відношень логічного set-наслідку маємо властивість монотонності, для них виконуються властивості деконпозиції формул $\neg \neg_L, \neg \neg_R, \vee_L, \vee_R, \neg \vee_L, \neg \vee_R, \neg_L, \neg_R$ (див. [2, 5])

Властивості відношень логічного set-наслідку, пов'язані з еквівалентними перетвореннями, базуються на властивостях формул, пов'язаних з композиціями реномінації. В $L_{\perp \perp}^Q$ ці властивості отримуємо на основі властивостей $R, R_{\perp I}, R_{\perp U}, R_{\uparrow 1}, R_{\uparrow 2}, R_{\perp \neg}, R_{\perp \vee}, R_{\perp R}, R_{\perp \exists s}, R_{\perp \exists}$; в L_1^Q такі властивості отримуємо на основі $R, RI, RU, R_{\perp \neg}, R_{\perp \vee}, RR, R_{\perp \exists s}, R_{\perp \exists}$. Кожна така властивість формул R^* продукує 4 відповідні властивості $R_{\perp *L}, R_{\perp *R}, \neg R_{\perp *L}, \neg R_{\perp *R}$ для відношення логічного set-наслідку, коли виділена формула чи її заперечення – у лівій чи правій частині цього відношення.

У використанні властивостей елімінації кванторів під реномінацією власти-

вості типів $R\exists s$ та $R\exists$ мають допоміжний характер і не належать до базових.

Подібним чином властивості $R_{\perp\downarrow v}$, $R_{\perp\downarrow t}$ в $L_{\perp\downarrow}^Q$ та властивості $R_{\downarrow v}$, $R_{\downarrow t}$ в L_{\downarrow}^Q продукують по 4 властивості спрощення за реномінації предикатів-індикаторів.

Для прикладу наведемо властивості, індуковані $R_{\perp\downarrow v}$:

$$\begin{aligned} R_{\perp\downarrow vL}) R_{\bar{x},\perp,\downarrow,y}^{\bar{v},\bar{u},z}(\downarrow z), \Gamma \models_* \Delta &\Leftrightarrow \downarrow y, \Gamma \models_* \Delta; \\ R_{\perp\downarrow vR}) \Gamma \models_* R_{\bar{x},\perp,\downarrow,y}^{\bar{v},\bar{u},z}(\downarrow z), \Delta &\Leftrightarrow \Gamma \models_* \downarrow y, \Delta; \\ \neg R_{\perp\downarrow vL}) \neg R_{\bar{x},\perp,\downarrow,y}^{\bar{v},\bar{u},z}(\downarrow z), \Gamma \models_* \Delta &\Leftrightarrow \neg \downarrow y, \Gamma \models_* \Delta; \\ \neg R_{\perp\downarrow vR}) \Gamma \models_* \neg R_{\bar{x},\perp,\downarrow,y}^{\bar{v},\bar{u},z}(\downarrow z), \Delta &\Leftrightarrow \Gamma \models_* \neg \downarrow y, \Delta. \end{aligned}$$

У $L_{\perp\downarrow}$ маємо властивості елімінації \neg для константних \perp -формул $R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z)$:

$$\begin{aligned} El_{\neg\perp L}) \neg R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z), \Gamma \models_* \Delta &\Leftrightarrow R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z), \Gamma \models_* \Delta; \\ El_{\neg\perp R}) \Gamma \models_* \neg R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z), \Delta &\Leftrightarrow \Gamma \models_* R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z), \Delta. \end{aligned}$$

Для \perp -формул $R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z)$ маємо властивості елімінації \perp -формул:

$$\begin{aligned} El_{R\downarrow F}) R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z), \Gamma^* \models_{*F} \Delta &\Leftrightarrow \Gamma^* \models_{*F} \Delta; \\ El_{R\downarrow T}) \Gamma^* \models_T \Delta, R_{\bar{x},\perp,\perp}^{\bar{v},\bar{u},z}(z) &\Leftrightarrow \Gamma^* \models_{*T} \Delta. \end{aligned}$$

У $L_{\downarrow\perp}$ та $L_{\perp\downarrow}$ маємо властивості елімінації p_T -формул $\downarrow z$ та p_F -формул $\neg \downarrow z$:

$$\begin{aligned} El_{\downarrow F}) \downarrow z, \Gamma^* \models_F \Delta &\Leftrightarrow \Gamma^* \models_F \Delta; \\ El_{\downarrow T}) \Gamma^* \models_T \Delta, \neg \downarrow z &\Leftrightarrow \Gamma^* \models_T \Delta. \end{aligned}$$

Опишемо похідні властивості відношень логічного set-наслідку, пов'язані з T -формулами $\exists x \downarrow x$ та F -формулами $\exists x \neg \downarrow x$.

Зняття \neg в $\neg \exists x \downarrow x$ та $\neg \exists x \neg \downarrow x$:

$$\begin{aligned} \neg \exists L) \neg \exists x \downarrow x, \Gamma \models_* \Delta &\Leftrightarrow \Gamma \models_* \Delta, \exists x \downarrow x; \\ \neg \exists \downarrow R) \Gamma \models_* \Delta, \neg \exists x \downarrow x &\Leftrightarrow \exists x \downarrow x, \Gamma \models_* \Delta; \\ \neg \exists \neg \downarrow L) \neg \exists x \neg \downarrow x, \Gamma \models_* \Delta &\Leftrightarrow \Gamma \models_* \Delta, \exists x \neg \downarrow x; \\ \neg \exists \neg \downarrow R) \Gamma \models_* \Delta, \neg \exists x \neg \downarrow x &\Leftrightarrow \exists x \neg \downarrow x, \Gamma \models_* \Delta. \end{aligned}$$

Елімінація $\exists x \downarrow x$ та $\exists x \neg \downarrow x$:

$$\begin{aligned} El_{\exists\downarrow}) \exists x \downarrow x, \Gamma \models_* \Delta &\Leftrightarrow \Gamma \models_* \Delta; \\ El_{\exists\neg\downarrow}) \Gamma \models_* \Delta, \exists x \neg \downarrow x &\Leftrightarrow \Gamma \models_* \Delta. \end{aligned}$$

Також маємо елімінацію $\downarrow z$ зі “свіжим” неістотним іменем z , якщо $\downarrow z$ – зліва у відношенні логічного set-наслідку. Якщо ж $\downarrow z$ – справа у цьому відношенні, то для F -наслідку така елімінація некоректна, адже гарантовано виконується $\Gamma \models_{*F} \Delta, \downarrow z$, проте не завжди $\Gamma \models_{*F} \Delta: \exists x \downarrow x \stackrel{R}{\neq}_F \exists x \neg \downarrow x$.

Отже, маємо такі похідні властивості елімінації $\downarrow z$ за “свіжим” неістотним z : $El_{\downarrow L}) \downarrow z, \Gamma \models_* \Delta \Leftrightarrow \Gamma \models_* \Delta$, якщо $z \in fu(\Gamma, \Delta)$;

$El_{\downarrow R}) \Gamma \models_{*T} \Delta, \downarrow z \Leftrightarrow \Gamma \models_{*T} \Delta$, якщо $z \in fu(\Gamma, \Delta)$.

Опишемо властивості елімінації кванторів в L_{\downarrow}^Q та $L_{\perp\downarrow}^Q$. Важливо, що тут маємо *різні* описи елімінації кванторів для відношень логічного T -наслідку (вживаємо $z \downarrow$) та F -наслідку (вживаємо $\neg z \downarrow$).

В $L_{\perp\downarrow}^Q$ маємо такі властивості елімінації кванторів (тут $*$ – $P_{\perp\downarrow}$ чи $R_{\perp\downarrow}$):

$$\begin{aligned} \exists R_{\perp\downarrow TL}) \text{ за умови } z \in fu(\Gamma, \Delta, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi)) \\ R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \Gamma^* \models_T \Delta &\Leftrightarrow R_{\bar{w},\perp,z}^{\bar{v},\bar{u},x}(\Phi), z, \Gamma^* \models_T \Delta; \\ \neg \exists R_{\perp\downarrow TR}) \text{ за умови } z \in fu(\Gamma, \Delta, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi)) \\ \Gamma^* \models_T \neg R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \Delta &\Leftrightarrow \Gamma, \downarrow z^* \models_T \neg R_{\bar{w},\perp,z}^{\bar{v},\bar{u},x}(\Phi), \Delta; \end{aligned}$$

$$\begin{aligned} \exists R_{\perp\downarrow VTR}) \Gamma, \downarrow y^* \models_T \Delta, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi) &\Leftrightarrow \\ \Leftrightarrow \Gamma, \downarrow y^* \models_T \Delta, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), R_{\bar{w},\perp,y}^{\bar{v},\bar{u},x}(\Phi); \\ \neg \exists R_{\perp\downarrow VTl}) \neg R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), y, \Gamma^* \models_T \Delta &\Leftrightarrow \\ \Leftrightarrow \neg R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \neg R_{\bar{w},\perp,y}^{\bar{v},\bar{u},x}(\Phi), y, \Gamma^* \models_T \Delta; \end{aligned}$$

$$\begin{aligned} \exists R_{\perp\downarrow FL}) \text{ за умови } z \in fu(\Gamma, \Delta, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi)) \\ R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \Gamma^* \models_F \Delta &\Leftrightarrow R_{\bar{w},\perp,z}^{\bar{v},\bar{u},x}(\Phi), \Gamma^* \models_F \Delta, \neg \downarrow z; \\ \neg \exists R_{\perp\downarrow FR}) \text{ за умови } z \in fu(\Gamma, \Delta, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi)) \\ \Gamma^* \models_F \neg R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \Delta &\Leftrightarrow \Gamma^* \models_F \neg R_{\bar{w},\perp,z}^{\bar{v},\bar{u},x}(\Phi), \neg z, \Delta; \end{aligned}$$

$$\begin{aligned} \exists R_{\perp\downarrow VFR}) \Gamma^* \models_F \Delta, \neg y, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi) &\Leftrightarrow \\ \Leftrightarrow \Gamma^* \models_F \Delta, \neg y, R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), R_{\bar{w},\perp,y}^{\bar{v},\bar{u},x}(\Phi); \\ \neg \exists R_{\perp\downarrow VFL}) \neg R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \Gamma^* \models_F \neg \downarrow y, \Delta &\Leftrightarrow \\ \Leftrightarrow \neg R_{\bar{w},\perp}^{\bar{v},\bar{u}}(\exists x \Phi), \neg R_{\bar{w},\perp,y}^{\bar{v},\bar{u},x}(\Phi), \Gamma^* \models_F \neg \downarrow y, \Delta. \end{aligned}$$

У випадку відношення $P_{\perp\downarrow} \models_{IR}$ для опису елімінації кванторів достатньо властивостей $\exists R_{\perp\downarrow TL}$ та $\exists R_{\perp\downarrow VTR}$.

В L_{\downarrow}^Q маємо аналогічні властивості елімінації кванторів $\exists R_{\downarrow TL}$, $\neg \exists R_{\downarrow TR}$, $\exists R_{\downarrow VTR}$, $\neg \exists R_{\downarrow VTl}$, $\exists R_{\downarrow FL}$, $\neg \exists R_{\downarrow FR}$, $\exists R_{\downarrow VFR}$, $\neg \exists R_{\downarrow VFL}$.

Відмінність описів елімінації кванторів для відношень логічного T -наслідку та логічного F -наслідку робить неможливим подібний опис одним рядком для відношень типу TF .

Допоміжні властивості \downarrow -розподілу та первісного означення в L_{\downarrow}^Q та $L_{\perp\downarrow}^Q$ теж різні для відношень типу T та типу F .

$$\begin{aligned} \text{Для відношень типу } T \text{ маємо:} \\ \downarrow vT) \Gamma^* \models_T \Delta &\Leftrightarrow \downarrow z, \Gamma^* \models_T \Delta; \\ \downarrow dT) \Gamma^* \models_T \Delta &\Leftrightarrow \Gamma^* \models_T \Delta, \downarrow y \text{ та } \downarrow y, \Gamma^* \models_T \Delta. \end{aligned}$$

$$\begin{aligned} \text{Для відношень типу } F \text{ маємо:} \\ \downarrow vF) \Gamma^* \models_F \Delta &\Leftrightarrow \Gamma^* \models_F \neg \downarrow z, \Delta \end{aligned}$$

$\downarrow dF) \Gamma^* \models_F \Delta \Leftrightarrow \Gamma^* \models_F \Delta, \neg \downarrow y$ та $\neg \downarrow y, \Gamma^* \models_F \Delta$.

Властивості $\downarrow vT$, $\downarrow dT$, $\downarrow vF$, $\downarrow dF$ також виконуються для відношень типу IR , водночас як базові властивості для цих відношень доцільно взяти простіші $\downarrow vT$ та $\downarrow dT$.

Опис в L_1^Q та $L_{1\perp}^Q$ властивостей відношень логічного наслідку типу TF зводиться до паралельного опису таких властивостей для відношень типу T та типу F .

Таким чином, в логіках L_1^Q та $L_{1\perp}^Q$ відношення типу TF мають ознаки девіантності, тому окремо розглядати ці відношення далі не будемо.

Опишемо тепер властивості, які *гарантовано* виконуються для відношень логічного set-наслідку певного типу.

Для усіх розглянутих відношень логічного set-наслідку маємо:

C) $\Phi, \Gamma \models^* \Delta, \Phi$.

Додатково гарантують наявність відношення певного типу такі властивості.

CL) $\Phi, \neg \Phi, \Gamma^P \models_T \Delta$;

CR) $\Gamma^P \models_F \Delta, \Phi, \neg \Phi$;

CLR) $\Phi, \neg \Phi, \Gamma^P \models_{TF} \Delta, \Psi, \neg \Psi$.

В логіці $L_{1\perp}^Q$ маємо властивості гарантованої наявності відношень логічного set-наслідку на базі \perp -формул $R_{\bar{x}, \perp, \perp}^{\bar{v}, \bar{u}, z}(z)$:

$C_{\perp L} R_{\bar{w}, \perp, \perp}^{\bar{v}, \bar{u}, x}(x), \Gamma^* \models_T \Delta$;

$C_{\perp IR} \Gamma^* \models_F \Delta, R_{\bar{w}, \perp, \perp}^{\bar{v}, \bar{u}, x}(x)$.

В $L_{1\perp}^Q$ й L_1^Q маємо властивості гарантованої наявності F -наслідку і T -наслідку на базі pT -формул $\downarrow z$ та pF -формул $\neg \downarrow z$:

C₁) $\Gamma^* \models_F \downarrow z, \Delta$;

C₋₁) $\neg \downarrow z, \Gamma, * \models_T \Delta$.

На основі описаних *властивостей* гарантованої наявності певного відношення $\Gamma \models^* \Delta$ отримуємо однойменні *умови* гарантованої наявності деякого $\Gamma \models^* \Delta$:

C) існує формула Φ : $\Phi \in \Gamma$ та $\Phi \in \Delta$;

CL) існує формула Φ : $\Phi \in \Gamma$ та $\neg \Phi \in \Gamma$;

CR) існує формула Φ : $\Phi \in \Delta$ та $\neg \Phi \in \Delta$;

$C_{\perp L}$) існує формула $R_{\bar{w}, \perp, \perp}^{\bar{v}, \bar{u}, x}(x) \in \Gamma$;

$C_{\perp IR}$) існує формула $R_{\bar{w}, \perp, \perp}^{\bar{v}, \bar{u}, x}(x) \in \Delta$;

C₁) існує формула $\downarrow x \in \Delta$;

C₋₁) існує формула $\neg \downarrow x \in \Gamma$.

Тепер для конкретного відношення логічного set-наслідку отримуємо наступні *умови* його гарантованої наявності:

$P_{\perp L} \models_{IR}$: умова $C \vee C_{\perp} \vee C_{\perp L} \vee C_{\perp IR}$;

$P_{\perp L} \models_T$: умова $C \vee C_L \vee C_{\perp} \vee C_{\perp L}$;

$P_{\perp L} \models_F$: умова $C \vee C_R \vee C_{\perp} \vee C_{\perp IR}$;

$R_{\perp L} \models_T$: умова $C \vee C_{\perp} \vee C_{\perp L}$;

$R_{\perp L} \models_F$: умова $C \vee C_{\perp} \vee C_{\perp IR}$;

$P_{\perp} \models_{IR}$: умова $C \vee C_{\perp}$;

$P_{\perp} \models_T$: умова $C \vee C_L \vee C_{\perp}$;

$P_{\perp} \models_F$: умова $C \vee C_R \vee C_{\perp}$;

$R_{\perp} \models_T$: умова $C \vee C_{\perp}$;

$R_{\perp} \models_F$: умова $C \vee C_{\perp}$.

Ці умови індукують умови замкненості секвенції в численні, яке формалізує відповідне відношення set-наслідку.

Підсумовуючи, для кожного з розглянутих відношень логічного set-наслідку типів IR , T , F , наведемо базові властивості, які продукують секвенційні форми відповідного числення. Для виразності ці властивості поєднаємо в групи, а потім укажемо такі групи для кожного з відношень.

Властивості декомпозиції формул:

$D_{IR} = \{\neg_L, \neg_R, \vee_L, \vee_R\}$;

$D_{TF} = \{\neg\neg_L, \neg\neg_R, \vee_L, \vee_R, \neg\vee_L, \neg\vee_R\}$.

Властивості елімінації кванторів:

$\exists_{LIR} = \{\exists R_{L\perp TL}, \exists R_{L\perp VTR}, \downarrow dT, \downarrow vT\}$;

$\exists_{\perp T} = \{\exists R_{L\perp TL}, \neg \exists R_{L\perp TR}, \exists R_{L\perp VTR}, \neg \exists R_{L\perp VTL}, \downarrow dT, \downarrow vT\}$;

$\exists_{\perp F} = \{\exists R_{L\perp FL}, \neg \exists R_{L\perp FR}, \exists R_{L\perp VFR}, \neg \exists R_{L\perp VFL}, \downarrow dF, \downarrow vF\}$;

$\exists_{IR} = \{\exists R_{L\perp TL}, \exists R_{L\perp VTR}, \downarrow dT, \downarrow vT\}$;

$\exists_T = \{\exists R_{L\perp TL}, \neg \exists R_{L\perp TR}, \exists R_{L\perp VTR}, \neg \exists R_{L\perp VTL}, \downarrow dT, \downarrow vT\}$;

$\exists_F = \{\exists R_{L\perp FL}, \neg \exists R_{L\perp FR}, \exists R_{L\perp VFR}, \neg \exists R_{L\perp VFL}, \downarrow dF, \downarrow vF\}$.

Властивості еквівалентних перетворень:

$R_{LIR} = \{R_L, R_R, R_{\perp L}, R_{\perp IR}, R_{\perp UL}, R_{\perp UR}, R_{\perp RL}, R_{\perp RR}, R_{\perp \neg L}, R_{\perp \neg R}, R_{\perp \vee L}, R_{\perp \vee R}\}$;

$R_{LTF} = \{R_L, R_R, \neg R_L, \neg R_R, R_{\perp L}, R_{\perp IR}, \neg R_{\perp L}, \neg R_{\perp IR}, R_{\perp UL}, R_{\perp UR}, \neg R_{\perp UL}, \neg R_{\perp UR}, R_{\perp RL}, R_{\perp RR}, \neg R_{\perp RL}, \neg R_{\perp RR}, R_{\perp \neg L}, R_{\perp \neg R}, \neg R_{\perp \neg L}, \neg R_{\perp \neg R}, R_{\perp \vee L}, R_{\perp \vee R}, \neg R_{\perp \vee L}, \neg R_{\perp \vee R}\}$;

$R_{IR} = \{R_L, R_R, R_L, R_R, R_U, R_U, R_R, R_R, R_{\neg L}, R_{\neg R}, R_{\vee L}, R_{\vee R}\}$;

$$R_{TF} = \{R_L, R_R, \neg R_L, \neg R_R, R_{IL}, R_{IR}, \neg R_{IL}, \neg R_{IR}, R_{UL}, R_{UR}, \neg R_{UL}, \neg R_{UR}, R_{RL}, R_{RR}, \neg R_{RL}, \neg R_{RR}, R_{\neg L}, R_{\neg R}, \neg R_{\neg L}, \neg R_{\neg R}, R_{\vee L}, R_{\vee R}, \neg R_{\vee L}, \neg R_{\vee R}\}.$$

Властивості, пов'язані з предикатами-індикаторами $\downarrow z$:

$$I_{LIR} = \{R_{\perp L \vee L}, R_{\perp L \vee R}, R_{\perp L \wedge L}, R_{\perp L \wedge R}; R_{\uparrow L \wedge L}, R_{\uparrow L \wedge R}, R_{\uparrow L \wedge L}, R_{\uparrow L \wedge R}; El_{R \wedge L}, El_{R \wedge R}; El_{L \wedge L}, El_{L \wedge R}\};$$

$$I_{LT} = I_{LB} \cup \{El_{R \wedge L}, El_{L \wedge L}\};$$

$$I_{LF} = I_{LB} \cup \{El_{R \wedge L}, El_{L \wedge L}\};$$

$$I_{IR} = \{R_{\perp L \vee L}, R_{\perp L \vee R}, R_{\perp L \wedge L}, R_{\perp L \wedge R}; El_{L \wedge L}\}.$$

$$I_T = I_B \cup \{El_{L \wedge L}\};$$

$$I_F = I_B \cup \{El_{L \wedge L}\}.$$

$$T_{UT} \quad I_{LB} = \{R_{\perp L \vee L}, R_{\perp L \vee R}, \neg R_{\perp L \vee L}, \neg R_{\perp L \vee R}, R_{\perp L \wedge L}, R_{\perp L \wedge R}, \neg R_{\perp L \wedge L}, \neg R_{\perp L \wedge R}; R_{\uparrow L \wedge L}, R_{\uparrow L \wedge R}, \neg R_{\uparrow L \wedge L}, \neg R_{\uparrow L \wedge R}, R_{\uparrow L \wedge L}, R_{\uparrow L \wedge R}, \neg R_{\uparrow L \wedge L}, \neg R_{\uparrow L \wedge R}; El_{\neg L}, El_{\neg R}\};$$

$$I_B = \{R_{L \vee L}, R_{L \vee R}, \neg R_{L \vee L}, \neg R_{L \vee R}, R_{L \wedge L}, R_{L \wedge R}, \neg R_{L \wedge L}, \neg R_{L \wedge R}\}.$$

Маємо такі групи базових властивостей для відношень логічного set-наслідку.

$$P_{\perp L} | =_{IR}: D_{IR} \cup R_{LIR} \cup I_{LIR} \cup \exists I_{IR};$$

$$P_{\perp L} | =_T \text{ та } R_{\perp L} | =_T: D_{TF} \cup R_{LTF} \cup I_{LTF} \cup \exists I_T;$$

$$P_{\perp L} | =_F \text{ та } R_{\perp L} | =_F: D_{TF} \cup R_{LTF} \cup I_{LTF} \cup \exists I_F;$$

$$P_{\perp L} | =_{IR}: D_{IR} \cup R_{LIR} \cup I_{LIR} \cup \exists I_{IR};$$

$$P_{\perp L} | =_T \text{ та } R_{\perp L} | =_T: D_{TF} \cup R_{LTF} \cup I_{LTF} \cup \exists I_T;$$

$$P_{\perp L} | =_F \text{ та } R_{\perp L} | =_F: D_{TF} \cup R_{LTF} \cup I_{LTF} \cup \exists I_F.$$

Зауважимо, що 4 пари відношень логічного set-наслідку мають *однакові* базові властивості; в цих парах відношення типу T та типу F відрізняються *різними* умовами гарантованої наявності.

Висновки

У роботі досліджено семантичні властивості нових класів програмно-орієнтованих логік часткових квазіарних предикатів без обмеження монотонності. Їхньою характерною особливістю є використання часткових предикатів-індикаторів, які визначають наявність у вхідних даних компоненти з відповідним предметним іменем, тобто з'ясовують означеність цього імені. Описано композиційні алгебри та мови цих логік, досліджено відношення логічного наслідку для формул та для множин формул мови. Розглянуто умови гарантованої наявності цих відношень, наведено їхні основні властивості. Особливу

увагу приділено властивостям, пов'язаним із предикатами-індикаторами, та властивостям елімінації кванторів. Властивості відношень логічного наслідку є семантичною основою побудови відповідних числень секвенційного типу, що буде зроблено в наступних роботах. Базові властивості певного відношення логічного наслідку продукують секвенційні форми відповідного числення, а умови гарантованої наявності такого відношення індукують умови замкненості секвенції в цьому численні.

Література

1. S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum (eds), Handbook of Logic in Computer Science, Vol. 1–5, Oxford University Press, 1993–2000.
2. М. С. Нікітченко, О. С. Шкільняк, С. С. Шкільняк, Чисті першопорядкові логіки квазіарних предикатів, Проблеми програмування, 2016, № 2–3, С. 73–86.
3. С. С. Шкільняк, Першопорядкові композиційно-номінативні логіки з предикатами слабкої та строгої рівності, Проблеми програмування, 2019, № 3, С. 28–44.
4. М. С. Нікітченко, О. С. Шкільняк, С. С. Шкільняк, Секвенційні числення першопорядкових логік часткових предикатів з розширеними реномінаціями та композицією предикатного доповнення, Проблеми програмування, 2020, № 2–3, С. 182–197.
5. O. Shkilniak, S. Shkilniak. First-Order Sequent Calculi of Logics of Quasiary Predicates with Extended Renominations and Equality, UkrPROG'2022, CEUR Workshop Proceedings (CEUR-WS.org), 2023, pp. 3–18.
6. S.C. Kleene, Mathematical Logic, Dower Publications, 2013.

References

1. S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum (eds), Handbook of Logic in Computer Science, Vol. 1–5, Oxford University Press, 1993–2000.
2. M. Nikitchenko, O. Shkilniak, S. Shkilniak, Pure first-order logics of quasiary predicates, in Problems in Programming, 2016, No 2–3. pp. 73–86.

3. S. Shkilniak, First-order composition-nominative logics with predicates of weak equality and of strong equality, in Problems in Programming, 2019, No 3, pp. 28–44.
4. M. Nikitchenko, O. Shkilniak, S. Shkilniak, Sequent Calculi of First-order Logics of Partial Predicates with Extended Renominations and Composition of Predicate Complement, in Problems in Programming, 2020, No 2–3, pp. 182–197.
5. O. Shkilniak, S. Shkilniak. First-Order Sequent Calculi of Logics of Quasiary Predicates with Extended Renominations and Equality, UkrPROG'2022, CEUR Workshop Proceedings (CEUR-WS.org), 2023, pp. 3–18.
6. S.C. Kleene, Mathematical Logic, Dower Publications, 2013.

Одержано: 06.09.2024

Внутрішня рецензія отримана: 13.09.2024

Зовнішня рецензія отримана: 16.09.2024

Про автора:

Шкільняк Степан Степанович,
доктор фізико-математичних наук,
професор,
<http://orcid.org/0000-0001-8624-5778>.

Місце роботи автора:

Київський національний університет
імені Тараса Шевченка,
Тел. (+38) (044) 521-33-45
E-mail: ss.sh@knu.ua,
Сайт: csc.knu.ua

А.Ю. Дорошенко, Я.О. Гайдукевич, В.О. Гайдукевич, О.С. Жиренков

КЛІЄНТО-ЦЕНТРИЧНИЙ ТЕХНОЛОГІЧНИЙ СТЕК ДЛЯ ПРОГНОЗУ ПОГОДИ ТА ЯКОСТІ ПОВІТРЯ

У роботі запропоновано багатоетапний процес доставки прогнозних даних на мобільний пристрій кінцевого користувача при метеорологічному прогнозуванні. Початковим завданням є збір історичних даних про погоду та забруднення повітря. Після цього наступні кроки повинні дозволити побудувати інфраструктуру прогнозування: модель вхідних та вихідних параметрів повинна бути визначена для задачі регресії, алгоритм машинного навчання має бути визначений та його гіперпараметри мають бути оптимізовані. Така модель прогнозування має бути збережена у файлі і використана для створення загальнодоступного веб-сервісу, а на мобільному пристрої має бути встановлений додаток, який би запитував прогнозні дані у сервісу та виводив би їх на екран у вигляді багатопараметричного графіка. Запропонована концепція розподіленого програмного рішення охоплює наступні три головні аспекти цієї архітектури: модель машинного навчання, контейнеризований веб-сервіс та додаток інтерфейсу користувача на мобільному пристрої. Бібліотеки Numpy та Pandas мови Python були використані для підготовки набору даних, бібліотека Scikit-learn і алгоритм Histogram gradient boosting були застосовані для створення регресійної моделі. Наступні технології були використані для побудови веб-сервісу: Docker, Kubernetes, FastAPI та BentoML. Для створення мобільної програми використовувалася платформа Google Flutter. Ключові слова: задача регресії, машинне навчання, прогнозування, контейнер, веб-сервіс, мобільний застосунок, Google Flutter, Bentoml, Mlops, мікросервісна архітектура.

A.Yu. Doroshenko, Y.O. Haidukevych, V.O. Haidukevych, O.S. Zhyrenkov

USER-CENTRIC TECHNOLOGY STACK FOR WEATHER AND AIR POLLUTION FORECASTING

This paper proposes the multi-stage process of delivering the forecast data to end-user mobile device. The initial task here is to collect the historical weather and air pollution data. After that, the following steps should allow to build the forecasting infrastructure: the input-output model should be defined for regression task, the machine learning algorithm should be trained and its hyperparameters should be optimized, this forecasting model should be serialized to file and it should be used to create the publicly available web-service, the mobile device should have the application installed that would be querying the forecast data from the service and would be displaying the multi-parameter chart on the screen.

The proposed concept of user-centric distributed application covers the following three pillars of this architecture: machine learning model, containerized web-service and user interface application on mobile device. The Python-based libraries Numpy and Pandas were used to prepare the dataset, the Scikit-learn library and Histogram gradient boosting algorithm were leveraged to build the machine learning model. Here are the technologies employed to build the web-service: Docker, Kubernetes, FastAPI and BentoML. The Google Flutter platform was used to build the application for mobile devices.

Key words: regression task, machine learning, forecasting, docker container, web service, mobile application, Google Flutter, Bentoml, Mlops, microservices.

Вступ

Задача прогнозування погоди була актуальною у всі часи. Однак у 21-му столітті екологічна складова стала особливо важливою через зміни клімату та військові ризики. Усі погодні характеристики та параметри забруднення повітря взаємопов'язані та впливають один на одного. Як відомо, збільшення парникових газів призводить до глобального потепління. Отже,

ефективний підхід до прогнозування повинен враховувати всі доступні параметри в одній моделі. Саме це і створює додаткові труднощі, оскільки більша кількість вхідних параметрів ускладнює модель, і не всі параметри однаково корисні для цілей прогнозування.

У даній роботі запропоновано багатоетапний процес доставки прогнозних да-

них на мобільний пристрій кінцевого користувача. Початковим завданням є збір історичних даних про погоду та забруднення повітря, після чого будується інфраструктура прогнозування: модель вхідних та вихідних параметрів повинна бути визначена для задачі регресії, алгоритм машинного навчання має бути визначений та його гіперпараметри мають бути оптимізовані. Така модель прогнозування зберігається у файлі і використовується для створення загальнодоступного веб-сервісу, а на мобільному пристрої встановлюється додаток, який виконує запит прогнозних даних до цього веб-сервісу та виводить їх на екран у вигляді багатопараметричного графіка.

Набір даних для навчання

Як погодні дані, так і показники забруднення повітря для міста Київ були отримані із сайту openweathermap.org. Зокрема, цей сервіс дозволяє отримати багато атмосферних характеристик для довільних GPS координат. Основні колонки цього набору даних показані на рис. 1. Дана таблиця містить погодинні дані та 33'863 записи в цілому, від 25-11-2020 до 05-10-2024.

Для реєстрації напрямку вітру зазвичай використовуються градуси. Але цей формат не є добрим для алгоритмів машинного навчання [1] через розривність представлення між 359° та 0°. Одним із популярних методів вирішення цієї проблеми є використання синусу та косинусу відповідного кута [2]. Ці колонки були розраховані з використанням алгоритму написаного на Python.

Забруднення повітря для всіх показників вимірюється у мікрограмах на кубічний метр ($\mu\text{g}/\text{m}^3$). Найбільшим забруднювачем є чадний газ через високу концентрацію.

Параметри LevelPM2 та LevelPM10 позначають пилове забруднення частинками до 2.5 та до 10 мікрметрів відповідно. Крім того, значення PM10 включає рівень PM2.5. Частинки PM2.5 шкідливі тим, що потрапляють безпосередньо в кров людини. Частинки більше ніж PM10 зазвичай відфільтровуються в дихальних шляхах та не потрапляють в легені.

Якість прогнозу можна суттєво покращити з використанням циклічних параметрів [2]. Очевидно, добові та річні цикли безпосередньо впливають на температуру повітря. Тижневі цикли пов'язані із забрудненням повітря, тому що багато підприємств не працюють на вихідних. Ці вісім параметрів також були згенеровані з використанням алгоритму, реалізованого на Python. Робота з фреймами даних була спрощена бібліотеками pandas та numpy.

Етап машинного навчання

Для навчання регресійної моделі були обрані наступні набори вхідних та вихідних параметрів:

```
input_columns = \
    ['SineDay', 'CosineDay',
     'SineWeek', 'CosineWeek',
     'SineMonth', 'CosineMonth',
     'SineYear', 'CosineYear',
     'Temperature', 'DewPoint',
     'Pressure', 'Humidity',
     'WindSpeed', 'WindSine',
     'WindCosine', 'CloudLevel',
     'LevelCO', 'LevelNO',
     'LevelNO2', 'LevelO3',
     'LevelSO2', 'LevelNH3',
     'LevelPM2', 'LevelPM10',
     'Temperature-M1', 'DewPoint-M1',
     'Pressure-M1', 'Humidity-M1',
     'WindSpeed-M1', 'WindSine-M1',
```

	A	B	C	D	E	F	G	H	N	O	P	Q	R	S	T	U
1	UtcTime	LocalDate	LocalHour	Temperature	DewPoint	Pressure	Humidity	WindSpeed	LevelCO	LevelNO	LevelNO2	LevelO3	LevelSO2	LevelNH3	LevelPM2	LevelPM10
2	1606266000	2020-11-25	3	3.29	1.01	1022	85	4.38	223.64	0	4.24	49.35	3.46	0.46	3.18	5.08
3	1606269600	2020-11-25	4	3.24	1.13	1021	86	4.23	226.97	0	4.33	48.64	3.73	0.47	3.1	4.89
4	1606273200	2020-11-25	5	3.43	1.79	1021	89	4.09	230.31	0	4.8	46.49	4.05	0.45	3.72	5.5
5	1606276800	2020-11-25	6	3.56	1.76	1021	88	0.45	233.65	0.01	5.83	43.63	4.59	0.48	4.49	6.33
6	1606280400	2020-11-25	7	3.76	1.96	1021	88	4.1	243.66	0.02	8.82	39.34	5.25	0.55	5.24	7.22
7	1606284000	2020-11-25	8	3.71	1.75	1021	87	4.44	250.34	0.04	11.48	36.48	5.84	0.59	5.97	8.04
8	1606287600	2020-11-25	9	3.77	1.81	1021	87	4.78	253.68	0.08	11.82	36.84	6.2	0.59	6.7	8.66
33860	1728154800	2024-10-05	22	15.27	14.96	1012	98	0.45	236.99	0	8.4	43.99	3.22	0.46	7.43	8.24
33861	1728158400	2024-10-05	23	15.07	14.76	1011	98	0.45	230.31	0	6.6	45.42	3.01	0.4	8.15	8.8
33862	1728162000	2024-10-06	0	15.07	14.76	1011	98	0.45	226.97	0	3.98	48.64	2.86	0.34	8.6	9.07
33863	1728165600	2024-10-06	1	15.17	14.86	1011	98	4.25	223.64	0	2.72	50.07	2.86	0.32	8.63	8.96
33864	1728169200	2024-10-06	2	15.08	14.77	1010	98	0.45	223.64	0	2.4	50.78	2.92	0.33	8.52	8.82

Рис. 1. Погодинна інформація про погодні умови та забруднення повітря, м. Київ.

```
'WindCosine-M1', 'CloudLevel-M1',
'LevelCO-M1', 'LevelNO-M1',
'LevelNO2-M1', 'LevelO3-M1',
'LevelSO2-M1', 'LevelNH3-M1',
'LevelPM2-M1', 'LevelPM10-M1']
output_columns = \
['Temperature-P1', 'DewPoint-P1',
'Pressure-P1', 'Humidity-P1',
'WindSpeed-P1', 'WindSine-P1',
'WindCosine-P1', 'CloudLevel-P1',
'LevelCO-P1', 'LevelNO-P1',
'LevelNO2-P1', 'LevelO3-P1',
'LevelSO2-P1', 'LevelNH3-P1',
'LevelPM2-P1', 'LevelPM10-P1']
```

Тут суфікс M1 позначає значення показника годину тому, а суфікс P1 – через годину. В подальшому модель даних може бути покращена, і глибина історії прогнозування може бути збільшена. Але дана модель вже дозволяє ітеративно прогнозування на довільний час наперед.

Набір даних був розділений на навчальну та тестову вибірку у пропорції 80% до 20% за допомогою бібліотечної функції `train_test_split`. Алгоритмом регресії було обрано `Histogram Gradient Boosting`, що є переможцем багатьох комерційних змагань. Слід відзначити швидкість цього алгоритму, що досягається за рахунок квантизації векторів навчальної вибірки. Наприклад, фаза тренування на поточному наборі даних потребувала лише 45 секунд. Натренована модель була збережена у файл у форматах `ONNX` та `Pickle` [3–5] для подальшого використання. Точність прогнозування на годину наперед наведена в табл. 1.

Таблиця 1

Точність прогнозування на годину наперед

Параметр	R2 Score	MAE
Temperature-P1	0.995264	0.436919
DewPoint-P1	0.981279	0.590434
Pressure-P1	0.995751	0.340160
Humidity-P1	0.935305	2.676903
WindSpeed-P1	0.146426	0.915481
WindSine-P1	0.491392	0.299113
WindCosine-P1	0.683517	0.234346
CloudLevel-P1	0.871486	7.293946
LevelCO-P1	0.977555	3.846348
LevelNO-P1	0.858722	0.444098

LevelNO2-P1	0.006143	2.336877
LevelO3-P1	0.974631	2.397414
LevelSO2-P1	0.931484	0.511058
LevelNH3-P1	0.926684	0.151170
LevelPM2-P1	0.969548	0.425212
LevelPM10-P1	0.946920	0.537251

Мікросервісний підхід до архітектури програмного забезпечення

У сучасний період розвитку штучного інтелекту (AI) та машинного навчання (ML) важливим є гарантування швидкої розробки та розгортання програмного забезпечення, яке інтегрує розроблені моделі та доводить їх до кінцевого споживача. Мікросервісна архітектура є важливим підходом, що спрощує цей процес завдяки декомпозиції систем на незалежні компоненти. Цей підхід дозволяє командам розробників працювати над різними частинами системи паралельно, що значно прискорює цикл розробки та впровадження нових функцій [6].

Мікросервісна архітектура базується на принципі декомпозиції системи на слабо пов'язані між собою компоненти, кожен з яких виконує окрему функцію. Це дозволяє забезпечувати:

- **Масштабованість (Scalability):** можливість незалежного горизонтального масштабування компонентів.
- **Гнучкість (Adaptability):** підвищення адаптивності системи, швидкості внесення змін як реакції на зміну бізнес-вимог.
- **Ремонтпридатність (Maintainability):** простота оновлення або заміни окремих сервісів.

MLOps підходи

MLOps – це практичний підхід, який поєднує машинне навчання (ML) та операційні процеси (Ops), спрямований на автоматизацію та оптимізацію життєвого циклу моделей машинного навчання. Цей підхід дозволяє командам швидше впроваджувати нові алгоритми та моделі, забезпечуючи безперервну інтеграцію і доставку (CI/CD) для проєктів машинного навчання [7].

У свою чергу це сприяє підвищенню якості моделей та зменшенню часу виходу на ринок нових продуктів, що є критично важливим у сучасному конкурентному середовищі [8].

На рис. 2 наведено діаграму Ейлера-Венна, яка описує логічне місце MLOps практик у комплексі задач, пов'язаних із машинним навчанням.

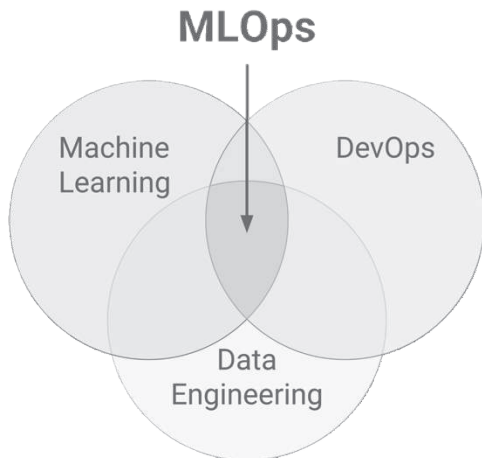


Рис. 2. Місце MLOps серед задач машинного навчання та інженерії програмного забезпечення

Прикладами MLOps фреймворків, які надають подібний інструментарій є:

- **MLFlow** – платформа з відкритим вихідним кодом, яка полегшує управління робочими процесами машинного навчання за допомогою експериментів відстеження, упаковки коду у відтворювані запуски та спільного використання моделей у середовищі співпраці;

- **KubeFlow** – платформа, що спрощує розгортання та управління робочими процесами машинного навчання на Kubernetes, забезпечуючи масштабованість і гнучкість для різноманітних проєктів.

- **BentoML** – платформа, яка дозволяє легко упаковувати та розгортати моделі машинного навчання у вигляді веб-сервісів, що спрощує інтеграцію з існуючими додатками та системами.

Також існує велика кількість комерційних MLOps систем, як-от AWS SageMaker, Microsoft Azure Machine Learning та інші, які не є предметом даного дослідження, але про які необхідно згадати.

У контексті MLOps мікросервіси сприяють:

- **Розподілу завдань:** команди можуть самостійно розробляти, тестувати та розгортати моделі.

- **Контейнеризації:** використання Docker для ізоляції моделей та їх залежностей. Контейнеризація спрощує процес розгортання та повторного використання моделей. Вона також підвищує ефективність управління залежностями, полегшуючи обслуговування.

- **Оркестрації:** автоматизація координації мікросервісів через інструменти, як Apache Airflow [9], Temporal [10], Dagster [11] та інші інструмента оркестрації роботи з даними засновані на формування циклу роботи з даними у вигляді спрямованих ациклічних графів (DAG – Direct Acyclic Graphs).

Запропонована архітектура

Для вирішення задачі побудови клієнто-центричного додатку прогнозування погодних умов з використанням моделей машинного навчання було використано підхід, що передбачає наступні структурні компоненти:

- **Сервіс машинного навчання.** Використано технологію BentoML – MLOps програмне забезпечення для розгортання масштабованих додатків та розміщення моделей будь-якого типу.

- **Серверна частина (Backend):** FastAPI – фреймворк, призначений для створення швидких та ефективних веб-додатків, які забезпечують REST API, гарантуючи водночас високу продуктивність та простоту інтеграції з іншими сервісами.

- **Оркестратор:** для експерименту було використано короткий python-скрипт, який забезпечує наповнення даних щодо прогнозованих значень погоди. У випадку подальшого розвитку проєкту за рахунок винесення оркестратора в окремий компонент його дуже легко замінити на більш масштабовані системи по оркестрації потоків роботи з даними.

Рис. 3 демонструє структуру компонентів системи, що забезпечують інтегра-

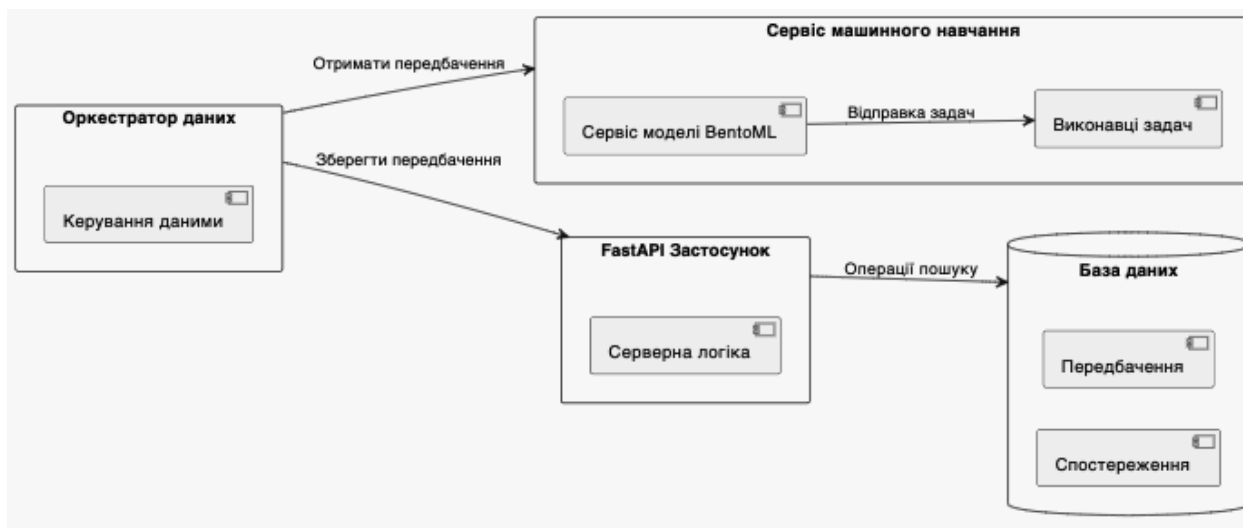


Рис. 3. Запропонована архітектура логіки серверної частини та роботи з моделлю машинного навчання.

цію між різними частинами сервісу прогнозування погоди.

У результаті було отримано набір точок доступу (рис. 4), які дозволяють здійснювати весь комплекс необхідних дій для забезпечення бізнес-логіки.

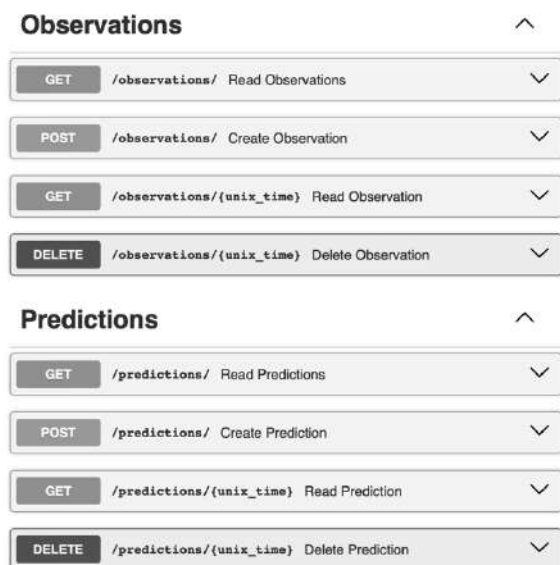


Рис. 4. Набір точок доступу до серверної логіки

Flutter та кросплатформність у створенні інтерактивного екологічного додатку

В основі розробки інтерактивного додатку для моніторингу погоди та якості повітря лежить **Flutter** — сучасний кросплатформний фреймворк, створений компанією Google. Завдяки своїм технічним можливостям Flutter дозволяє створювати

додатки з єдиним кодовим базисом, які працюють на Android, iOS, Web та Desktop. Це забезпечує значну економію часу та ресурсів, що є критично важливим у наукових проєктах і стартапах.

Основні переваги Flutter для кросплатформних проєктів:

1. **Єдиний код для всіх платформ:** Flutter дозволяє використовувати одну кодову базу для створення додатку на різних операційних системах, що спрощує його підтримку та оновлення.
2. **Гнучка система компонентів:** Завдяки розгалуженій бібліотеці компонентів (віджетів), включно з унікальними елементами UI, такими як графіки, картографічні компоненти та елементи взаємодії, можна швидко створювати складні інтерфейси.
3. **Швидкість розробки:** Функція "Hot Reload" дає змогу миттєво побачити зміни в інтерфейсі після редагування коду, що суттєво пришвидшує процес розробки.
4. **Продуктивність:** Flutter використовує власне обчислення зображення (без посередників на зразок JavaScript), що забезпечує високу швидкість і плавність роботи додатку навіть на пристроях із низькою продуктивністю.
5. **Широка екосистема бібліотек:** У проєкті використовується ряд потуж-

жних бібліотек, які допомагають реалізувати інтерактивність, обробку даних та адаптивний інтерфейс. Наприклад:

- I. **Syncfusion Flutter Charts** для візуалізації даних.
- II. **fl_chart** та **chart_sparkline** для створення графіків і лінійних графіків (sparkline).
- III. **flutter_map** і **latlong2** для інтеграції картографії.

Практичне застосування Flutter у проєкті:

В проєкті Flutter для створення інтерактивної екосистеми використовується компоненти, які дозволяють:

1. Відображати погодні параметри та індекс якості повітря (AQI) на карті України в реальному часі.
2. Інтегрувати графіки та візуалізації для аналізу змін у параметрах довкілля.
3. Забезпечити доступність додатка як на мобільних пристроях, так і у веб-браузерах.

Створення інтерактивної системи моніторингу погоди та якості повітря

Інтеграція сучасних технологій у сферу моніторингу стану довкілля є важливим етапом розвитку екологічної свідомості суспільства. Інтерактивні додатки, що поєднують дані про погоду та якість повітря, не лише забезпечують зручний доступ до інформації, а й сприяють глибшому розумінню впливу екологічних факторів на здоров'я людини. Одним із таких рішень є додаток, розроблений на базі Flutter, який дозволяє в реальному часі оцінювати екологічний стан ключових регіонів України.

Головна мета розробки такого додатку полягає в тому, щоб забезпечити інтерактивний доступ до екологічної інформації через карту, яка відображає погодні умови (температура, вологість, швидкість вітру тощо) та рівень забруднення повітря у вигляді індексу якості повітря (AQI). Візуальна складова, а саме кольорові маркери на карті, є важливим елементом дизайну, що сприяє швидкому сприйняттю інформації.

Для кожного з індексів AQI визначено колір, який символізує рівень якості повітря: зелений означає "добре", жовтий — "задовільно", помаранчевий — "помірно", червоний — "погано", а темно-коричневий — "дуже погано". Перевага такого підходу полягає в тому, що кольори дозволяють користувачеві миттєво оцінити ситуацію.

Дані для відображення отримуються з API OpenWeatherMap та власним API, в основі якої лежить модель машинного навчання для прогнозу метеорологічних даних. Сервіс надає вичерпну інформацію про погоду та стан атмосфери у форматі JSON. Приклад показників для міста Київ може включати такі параметри: температура — 18°C, тиск — 1015 гПа, вологість — 60%, швидкість вітру — 5 м/с, а також індекс AQI. Усі ці дані проходять обробку та візуалізацію в реальному часі.

Особливістю додатку є його інтерактивність. Наприклад, натискання на маркер міста відкриває вікно з детальною інформацією, де зазначаються погодні параметри та пояснення щодо рівня AQI. Це дозволяє користувачам не лише сприймати сухі цифри, а й отримувати пояснення щодо їх впливу. Окрім того, вбудовані графіки дають змогу аналізувати зміни певних параметрів протягом доби чи тижня.

Розробка таких додатків має значний потенціал для суспільного блага. З одного боку, вони підвищують екологічну обізнаність населення, а з іншого — можуть бути корисними для спеціалістів: екологів, метеорологів, міських планувальників. Наприклад, дані про якість повітря в конкретному регіоні можуть бути використані для визначення впливу промислових викидів або транспортного навантаження.

Отож, інтерактивний додаток для моніторингу погоди та якості повітря є сучасним інструментом, який сприяє зручному доступу до важливої екологічної інформації. Завдяки об'єднанню актуальних даних, інтерактивних візуалізацій і привабливого дизайну проєкт робить внесок у побудову свідомого ставлення до екології та здоров'я населення.

Візуалізація даних за допомогою інтерактивних графіків

Одним із ключових елементів інтерактивного додатку є графічне представлення прогнозів, яке дозволяє користувачам не лише отримувати поточні дані, а й аналізувати їхні зміни у часі (див. рис. 5). Завдяки інтеграції з API OpenWeatherMap та іншими метеорологічними сервісами, додаток пропонує прогнозування важливих параметрів, таких як температура, вологість, швидкість вітру, атмосферний тиск та рівень забруднення повітря (AQI) на кілька днів наперед.

Особливу увагу приділено візуалізації графіків, які дають змогу чітко простежити динаміку зміни погодних умов та рівня забруднення повітря протягом доби, тижня чи навіть місяця. Графічне подання цих даних не лише робить їх легшими для сприйняття, а й допомагає користувачам передбачати можливі зміни в екологічній ситуації. Наприклад, на графіках можна побачити, як змінюється температура та рівень забруднення повітря в певному регіоні залежно від часу доби або тижня, що дозволяє оцінити ймовірність виникнення шкідливих екологічних умов, таких як високий рівень AQI в певні періоди часу.



Рис. 5. Візуалізація показників AQI та метеорологічних даних у реальному часі

Графіки прогнозів можуть бути представлені як лінійні діаграми (рис. 6), що з'єднують значення певних параметрів на різних часових інтервалах, так і як гістограми, що дозволяють більш наочно від-

ображати порівняння між різними параметрами, як-от, рівнем забруднення на різних локаціях. Це дає змогу швидко зрозуміти, де саме і коли рівень забруднення може бути найбільш небезпечним для здоров'я людей.

Завдяки інтерактивним графікам, користувач може взаємодіяти з даними, обираючи різні періоди часу для аналізу та отримуючи відображення зміни погодних умов чи AQI за вибраний інтервал. Це дозволяє не тільки спостерігати за поточною ситуацією, а й робити прогнози на майбутнє, враховуючи сезонні чи погодні тенденції. Важливою перевагою є також можливість адаптувати графіки під різні типи даних, що робить додаток корисним як для простих користувачів, так і для професіоналів, таких як екологи та метеорологи, які можуть детальніше аналізувати прогнози для конкретних регіонів.

Таким чином, графічні прогнози у додатку не тільки сприяють кращому розумінню екологічної ситуації, а й допомагають в ухваленні обґрунтованих рішень щодо захисту здоров'я та навколишнього середовища.

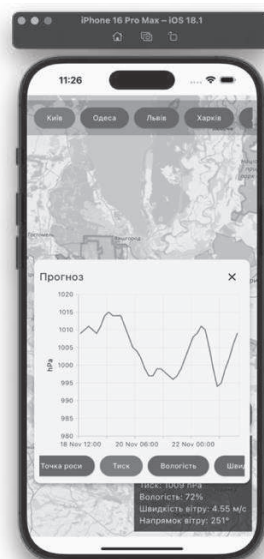


Рис. 6. Графічний прогноз метеорологічних даних на основі машинного навчання

Бібліотека **Syncfusion Flutter Charts** є потужним інструментом для візуалізації великої кількості даних у вигляді графіків, що допомагає здійснювати глибокий аналіз й ухвалювати обґрунтовані

рішення. Вона дозволяє інтегруватися з різноманітними джерелами даних, зокрема, через API, що робить її ідеальним вибором для прогнозування і моніторингу таких параметрів, як погодні умови, рівень забруднення повітря, чи навіть для аналізу економічних показників.

Такі інструменти використовуються в дослідженнях, де необхідно інтерактивно демонструвати дані з часом, виявляти тренди та аномалії, що є важливими у плануванні міської екологічної політики, прогнозуванні погоди та багатьох інших галузях науки та технологій.

У представленому коді (рис. 7), використовується компонент **SfCartesianChart** для створення лінійного графіку, який відображає дані про зміну певного параметра, наприклад, температури чи рівня забруднення повітря. Графік інтерактивний і дає можливість користувачу взаємодіяти з даними для кращого розуміння та прогнозування.

Це здійснюється за допомогою таких параметрів:

- **tooltipBehavior**: дозволяє показувати підказки, які з'являються при наведенні на точку графіка, що покращує взаємодію з даними;

- **primaryXAxis** та **primaryYAxis**: визначають осі графіка. Ось X використовує категоріальну шкалу для часу, де кожна мітка є датою та часом. За допомогою **AxisLabelFormatter** можна налаштувати відображення дат у зручному форматі (dd MMM HH:mm), що полегшує читання графіка;

- **LineSeries**: це основний тип ряду для лінійних графіків, який приймає дані про час та значення параметрів (наприклад, температура чи AQI). Важливим є те, що параметри даних вказуються як **Map<String, dynamic>**, що дає гнучкість у роботі з різними типами даних.

У цьому випадку дані про час і значення параметрів зберігаються у форматі словника (chartData), де для кожної точки графіка визначено, коли було отримано значення і яке саме значення параметра для цього часу.

```
return SfCartesianChart(
  tooltipBehavior: _tooltipBehavior,
  primaryXAxis: CategoryAxis(
    axisLabelFormatter: (AxisLabelRenderDetails details) {
      DateTime date = DateTime.parse(formattedString: details.text);
      return ChartAxisLabel(
        text: DateFormat(newPattern: 'dd MMM HH:mm').format(date: date), textStyle: details.textStyle);
    },
  ), // CategoryAxis
  primaryYAxis: NumericAxis(
    title: AxisTitle(
      text: yAxisLabel,
      textStyle: const TextStyle(color: Colors.black, fontSize: 14),
    ), // AxisTitle
  ), // NumericAxis
  series: <CartesianSeries<dynamic, dynamic>>[
    LineSeries<Map<String, dynamic>, String>(
      name: selectedParam,
      dataSource: chartData,
      xValueMapper: (Map<String, dynamic> data, int _) => data['time'],
      yValueMapper: (Map<String, dynamic> data, int _) => data[selectedParam],
      color: Colors.green,
    ), // LineSeries
  ], // <CartesianSeries>[]
); // SfCartesianChart
}
```

Рис. 7. Прогнозування погодних умов за допомогою бібліотеки Syncfusion Flutter Charts

Висновки

В роботі розроблена концепція розподіленого програмного рішення, орієнтованого на користувача, що охоплює наступні три головні аспекти запропонованої архітектури: модель машинного навчання, контейнеризований веб-сервіс та додаток інтерфейсу користувача на мобільному пристрої. Бібліотеки NumPy та Pandas мови Python були використані для підготовки набору даних, а бібліотека Scikit-learn і алгоритм Histogram gradient boosting були застосовані для створення регресійної моделі. Для побудови веб-сервісу були використані технології: Docker, Kubernetes, FastAPI та BentoML. Для створення мобільної програми використовувалася платформа Google Flutter.

References

1. C. M. Bishop, Pattern recognition and machine learning, Springer, 2006. [cited 04.03.2024]. <https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf>
2. A. Van Wyk, Encoding Cyclical Features for Deep Learning. [cited 04.03.2024]. <https://www.kaggle.com/code/avanwyk/encoding-cyclical-features-for-deep-learning>
3. Scikit-learn: Machine Learning in Python. [accessed 04.03.2024]. <https://scikit-learn.org/stable/>
4. Feature selection with scikit-learn library. [cited 04.03.2024]. https://scikit-learn.org/stable/modules/feature_selection.html
5. X. Dupre, O. Grisel, Accelerate and simplify Scikit-learn model inference with ONNX Runtime. [cited 04.03.2024]. <https://cloud-blogs.microsoft.com/open-source/2020/12/17/accelerate-simplify-scikit-learn-model-inference-onnx-runtime/>
6. Ribeiro, J. L., et al. "A Microservice Based Architecture Topology for Machine Learning Deployment." IEEE International Smart Cities Conference, Oct. 2019, <https://doi.org/10.1109/ISC246665.2019.9071708>.
7. Kreuzberger, Dominik, et al. Machine Learning Operations MLOps Overview Definition and Architecture.
8. Kim, Chorwon, et al. A Microservice-Based MLOps Platform for Efficient Development of AI Services in an Edge-Cloud Environment. Oct. 2023, <https://doi.org/10.1109/ictc58733.2023.10392296>.
9. Apache Airflow use cases [cited 20.11.2024] <https://airflow.apache.org/use-cases/>
10. Temporal IO use cases [cited 20.11.2024] <https://temporal.io/in-use>
11. Dagster use cases [cited 20.11.2024] <https://docs.dagster.io/getting-started>

Одержано: 29.11.2024

Внутрішня рецензія отримана: 04.12.2024

Зовнішня рецензія отримана: 06.12.2024

Про авторів:

^{1,2}Дорошенко Анатолій Юхимович, доктор фізико-математичних наук, професор, завідувач відділу <http://orcid.org/0000-0002-8435-1451>.

¹Гайдукевич Ярослав Олегович, аспірант. <http://orcid.org/0000-0002-6300-1778>.

¹Гайдукевич Владислав Олегович, аспірант. <http://orcid.org/0000-0002-0614-6778>.

¹Жиренков Олексій Сергійович, аспірант. <http://orcid.org/0009-0007-3124-1359>.

Місце роботи авторів:

¹ Інститут програмних систем
НАН України,
тел. +38-044-526-60-33
E-mail: a-y-doroshenko@ukr.net,
yarmcfly@gmail.com,
gaidukevichvlad@gmail.com,
ozhyrenkov@gmail.com

² НТУ України «Київський політехнічний інститут імені Ігоря Сікорського», факультет інформатики та обчислювальної техніки, тел. +38-044-204-86-10.

М. С. Єфремов, Ю. В. Крак

ПРОГРАМНЕ РІШЕННЯ ДЛЯ ЗБЕРІГАННЯ ТА АНАЛІЗУ ЕЛЕКТРОКАРДІОГРАМ

Розробка програмних рішень для роботи у медичній сфері з кожним роком стає популярнішою завдяки розвитку інформаційних технологій. За останні десятиліття алгоритми знаходження окремих ділянок на електрокардіограмі (ЕКГ), виявлення та класифікація аритмій та інших патологій на ЕКГ досягли значної точності. У даній роботі запропоновано підхід до інтеграції таких алгоритмів у єдину систему шляхом розробки програмного рішення, яке аналізує наявні ЕКГ з різних джерел, приводить їх до спільного формату та надає звітність щодо пацієнтів. Реалізація здійснена у вигляді веб-застосунку, що може надати користувачеві інформацію із кардіограм, обробляти ЕКГ залежно від доданих алгоритмів, зберігати потрібні ділянки ЕКГ, результати діагностування захворювань тощо. В систему імplementовано новий алгоритм знаходження R-піків та візуалізація ЕКГ-сигналу обробленим алгоритмом з анотаціями, отриманими як автоматично, так і за участі спеціалістів-кардіологів.

Ключові слова: електрокардіограма, оброблення сигналів, програмні рішення, база даних, формати даних

M.S. Yefremov, Yu.V. Krak

SOFTWARE SOLUTION FOR STORING AND ANALYZING ELECTROCARDIOGRAMS

Software solutions for medical and healthcare fields are becoming increasingly popular due to advancements in information technology. Over the past decades peak detection algorithms as well as detection of other segments on ECG, arrhythmias detection and other pathologies detection algorithms achieved significant accuracy. This study proposes an approach for integrating such algorithms into a unified systems by developing a software solution that analyzes available ECGs from various sources, standardizes them into a common format, and provides detailed reporting that is easily understandable for medical experts or patients. The implementation is carried out as a web application that can provide users with information about cardiograms, process ECGs depending on the integrated algorithms, store necessary ECG segments, diagnostic results, and more. The system incorporates a newly developed algorithm for detecting R-peaks and visualizing ECG signals processed by the algorithm with annotations obtained both automatically and with the participation of cardiology specialists.

Key words: electrocardiogram, signal processing, software solutions, database, data formats.

Вступ

Електрокардіограма є важливим інструментом діагностики роботи серця людини, на основі якого можна встановити діагноз виявлених захворювань серця та визначити профілактику кардіологічних захворювань. Аналіз ЕКГ дозволяє лікарям виявляти різноманітні серцеві

проблеми, такі як аритмії різного рівня, ішемії та інші порушення серцевого ритму. У контексті ЕКГ для аналізу сигналу використовують алгоритми автоматичного знаходження піків, (наприклад, R-піків), інтервалів (P-QRS-T) та інших характеристик ЕКГ (див. рисунок 1).

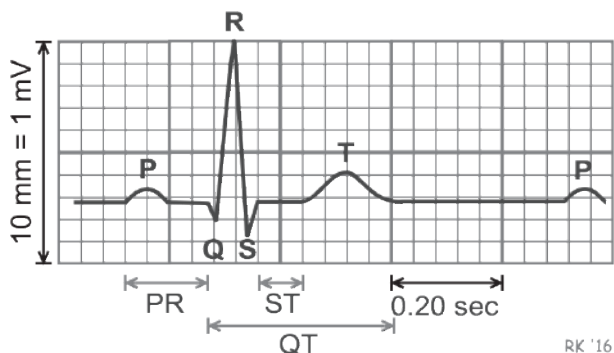


Рис. 1. Типовий аналіз одного серцевого ритму

Для оцінки точності та надійності цих алгоритмів необхідні великі та добре анотовані бази даних ЕКГ. Такі бази даних забезпечують стандартизовані набори даних, що включають різні типи серцевих ритмів і патологій, які можна використовувати для навчання та тестування алгоритмів. Такі бази стали незамінними ресурсами для вчених та розробників медичних програмних рішень.

Метою цієї статті є розроблення способів інтегрування алгоритмів обробки ЕКГ у системи, які можна використовувати у медичних закладах для аналізу та моніторингу станів пацієнтів. Для того, аби алгоритми аналізу працювали, треба щоб дані з холтерів або звичайних апаратів зняття ЕКГ можна було зчитати та перетворити у математичні моделі, зручні для роботи з алгоритмами. Відзначимо, що різні пристрої зняття ЕКГ-сигналів мають свої влаштовані формати подання даних, і приведення цих даних до певних стандартизованих форматів є надзвичайно важливою проблемою.

1. Огляд існуючих та стандартизованих форматів зберігання кардіограм

Для того, щоб здійснювати обробку ЕКГ та розробку застосунків, які працюють із кардіограмами важливо зрозуміти, в яких форматах зберігаються кардіограми, та які з них масово використовуються спеціалістами. Протягом останніх десятиліть дані з різних холтерів та кардіографів було збережено у базах даних, які мають відкритий доступ. Звернувшись до найпопулярніших

з таких баз, можна зробити висновки, які формати широко використовуються для зберігання і дослідження ЕКГ-сигналів. Наприклад, такі бази можна знайти на ресурсах Kaggle та PhysioNet [1]. Ці бази не просто мають вільний доступ, а були зібрані з кардіограм знятих з реальних пацієнтів. Розглянемо деякі бази із цих ресурсів. На ресурсі PhysioNet знаходиться велика кількість баз ЕКГ [1].

З основних можна виділити:

- MIT-BIH Arrhythmia Database[1].

Це одна з найвідоміших баз даних для аналізу серцевих аритмій, яка використовується у більшості досліджень, оскільки має анотовані сигнали різних випадків серцевих аритмій;

- European ST-T Database[1].

Ця база даних створена для оцінки алгоритмів виявлення та аналізу ST-сегменту і Т-хвилі. Вона містить записи від пацієнтів з різними серцевими станами;

- QT Database[1].

Вона містить записи ЕКГ з детальними анотаціями інтервалу QT, що важливо для дослідження інтервалу QT і пов'язаних з ним серцевих умов;

- Physikalisch-Technische Bundesanstalt (PTB) Diagnostic ECG Database[1]

Ця база даних містить багатоканальні записи ЕКГ від пацієнтів з різними серцевими захворюваннями, зокрема, інфарктом міокарда.

Що ж до ресурсу Kaggle, то на ньому є кілька баз даних ЕКГ, які використовуються, у тому числі як основні дані для проведення змагань для стимулювання розробки алгоритмів для автоматичного аналізу ЕКГ. База даних включає записи ЕКГ від китайських пацієнтів з різними типами аритмій. Однією з таких баз є база даних

- China Physiological Signal Challenge (CPSC) 2018[2].

Ця база даних містить записи ЕКГ з анотаціями, які можна використовувати для розробки і тестування алгоритмів розпізнавання аритмій і яка була наповнена 2018 року.

Наведемо інші бази з ресурсу Kaggle:

- China Physiological Signal Challenge (CPSC) 2020 Database[3].

Ця база даних була використана у змаганні CPSC 2020 і містить ширший набір даних для аналізу ЕКГ.

- Xinjiang Medical University ECG Data.

Ця база даних містить записи ЕКГ від пацієнтів з різними серцевими станами, надані Сінцзянським медичним університетом [4]

Для пошуку цих баз даних на Kaggle можна скористатися пошуком на платформі, ввівши відповідні ключові слова, наприклад, "ECG", "China", "CPSC". Kaggle також надає інструменти для роботи з даними та проведення змагань, що може бути корисним для тестування і розробки алгоритмів.

Розглянувши відкриті бази з сигналами ЕКГ, можна розділити файли сигналів ЕКГ на три категорії:

1. Файли у форматах стандартизованих ААМІ [5] такі як edf, bdf[6] тощо.
2. Бінарні файли у форматах, які були створені різними дослідницькими інститутами під себе, але, враховуючи їхню популярність, більшість програмних рішень, як-от wfdb[7], MATLAB мають бібліотеки та інструменти для роботи з ними.
3. Файли, які ніяк не шифруються, зазвичай це ЕКГ, зняті за короткий проміжок часу.

2. Декодування файлів ЕКГ та приведення до загального формату

Під час створення застосунку для аналізу кардіограм з метою його ширшого використання як у наукових дослідженнях, так і у прикладних застосуваннях, потрібно врахувати сумісність з якомога більшою кількістю форматів сигналу. Відзначимо, що, хоча описані вище формати й часто використовуються, підтримка будь-якого конкретного з них не гарантується у кардіографах та холтерах, особливо в пристроях, створених до стандартизації ЕКГ. Також постає проблема, пов'язана з тим, що здебільшого прилади, які повертають результати вимірювання, закодовані так, що їх можна прочитати лише використовуючи застосунки, розроблені авторами пристрою. Наприклад, холтер від фірми Philips[8], є комерційним продуктом, який часто використовується у медичних закладах, тому його формат .zhr[8] є поширеним та навіть підтримується wfdb. Навіть застосунок який іде разом з холтером має можливість конвертувати цей формат у стандартизовані edf, MIT тощо. Але розробники не надають файл метаданих, в якому зберігається інформація про розшифрування сигналу, тому постає проблема використання цих даних для розробників інших програмних засобів. Одним із способів вирішення цієї проблеми є метод підбору. Оскільки всі файли з кардіографів кодуються у бінарний формат, то підібрати функцію, що його декодує можливо, хоча ця процедура займає багато часу і під час декодування можливі втрати фрагментів сигналу. Так, під час роботи з холтером SDM-23 була спроба самостійно декодувати сигнал. Як результат сигнал втрачав деякі фрагменти (див. рисунок 2). Це зумовлено двома факторами: по-перше, відсутні метадані, де вказано параметри сигналу, які методом підбору не вийшло підібрати; по-друге, різні частини сигналу можуть бути закодовані різними методами. В результаті, окрім підбору алгоритму декодування, є потреба перебирати частинами, що є недоцільною витратою часу, оскільки заздалегідь невідомо, які саме методи та варіанти були закодовані.

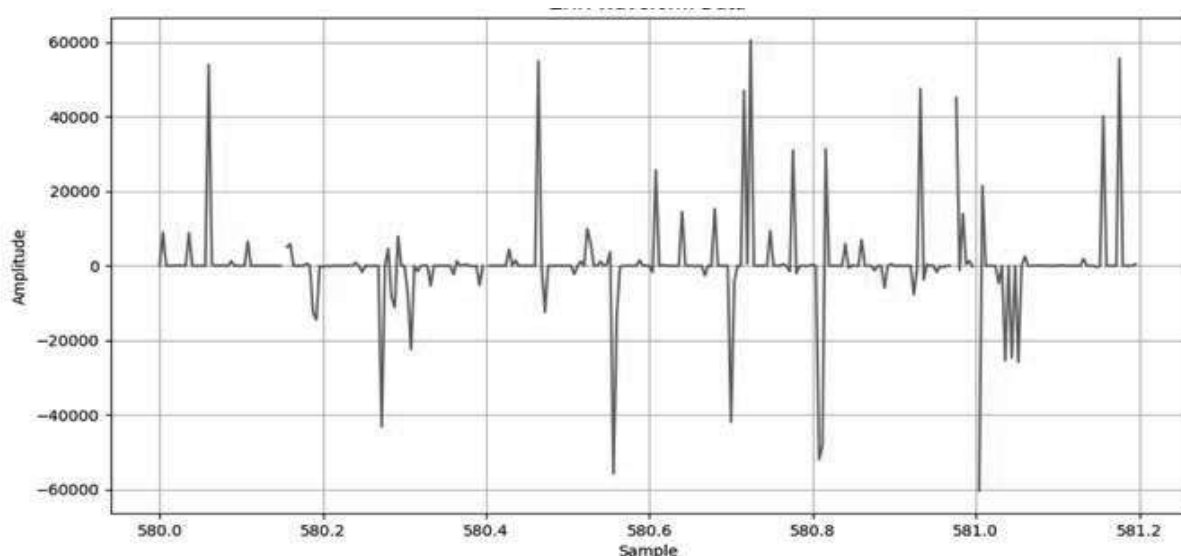


Рис 2. Невдале декодування сигналу ЕКГ

У таких випадках, коли сигнал неможливо якісно декодувати – єдиний вихід є напряму звертатись до розробників апаратури, або працівників медичних закладів та, за можливості, домовлятися про співпрацю.

Таким чином, з наведеної інформації можна зробити висновки, що під час обробки та аналізу ЕКГ даних у різних форматах важливою є розробка застосунку, який давав можливість враховувати ці формати і декодувати інформацію у доступні формати. Зважаючи на це було підібрано архітектуру, яка дозволяє використовувати формати розробки як доступні та які широко використовуються, так і формати локальних пристроїв. Використавши паттерн-програмування «фабрика» створюється контролер форматів, який буде відповідати за модулі для кожного нового формату, підтримка якого додаватиметься до системи.

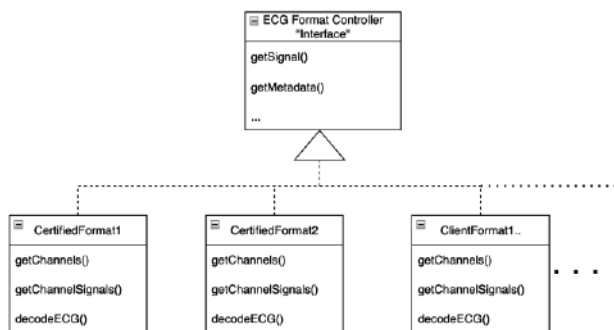


Рис 3. Фабрика декодування форматів ЕКГ

На рисунку 3 зображено реалізацію фабричного методу для декодування ЕКГ.

Основними компонентами для системи, що відповідає за декодування ЕКГ форматів, є контролер, який залежно від формату повертає з фабрики класів обробників потрібний обробник форматів. Контролер форматів визначає тип вхідних даних і направляє їх до відповідного обробника, здатного обробити конкретний формат. Фабрика обробників створює правильний парсер для даного формату, що витягує числові канали та метадані з вхідних даних, а також може додавати додаткову інформацію. Фабрика обробників є ключовою, оскільки вона забезпечує централізовану точку управління для створення обробників. Це дозволяє розширювати систему, додаючи нові формати без необхідності змінювати існуючий код, що підвищує модульність і підтримувальність (maintainability) системи. Кожен формат обробляється окремим обробником, що спрощує тестування та налагодження. Такий підхід особливо корисний у лікарнях, де використовуються власні формати ЕКГ, оскільки він дозволяє інтегрувати нові формати без значних змін у системі.

3. Архітектура рішення для роботи

Для повноцінної роботи системи, що здійснює обробку даних, пов'язаних з медициною варто враховувати, що такі дані містять конфіденційну інформацію про пацієнтів, яка охороняється законами про захист персональних даних, такими як GDPR у Європі [9] або HIPAA у США [10]. Забезпечення анонімності даних та їхній захист від несанкціонованого доступу є важливим, але складним завданням. Щоб не зберігати персональні дані клієнта, у запропонованому застосунку для роботи з ЕКГ, будемо підключати до нього реляційну базу даних, яка слугуватиме сховищем деанонізованим файлам. У цій базі будемо зберігати як закодовані файли, так й анонімізовані декодовані для подальшої їх обробки. Також до бази можна буде зберігати файли як через безпосередні застосунки поза даним сервісом, так й через

певний інтерфейс для користувачів цього застосунку.

Декодовані файли, без метаданих пацієнта вже можна передавати для їхньої подальшої обробки алгоритмами ЕКГ-аналізу. Так, дані з холтерів та кардіографів, які були отримані сервісом управління форматами, передаються на аналіз іншому сервісу, в якому реалізовані алгоритми обробки ЕКГ. Відзначимо, що був імплементований сервіс із знаходження R-піків, описаний у роботах [11,12] Цей алгоритм був протестований на вищезазначеній базі, а саме на базі з ресурсу [1], що підтвердило його ефективність та точність. Повна архітектура з усіх сервісів, які наразі доступні, можна побачити на рисунку 4. Така архітектура є досить гнучкою, та дозволяє інтегрувати його до різних лікарень без суттєвих проблем.

Для того, щоб будь-яка система мала успіх – важливою частиною є ті сервіси системи, які напряду комунікують із користувачем. У даному випадку - це є

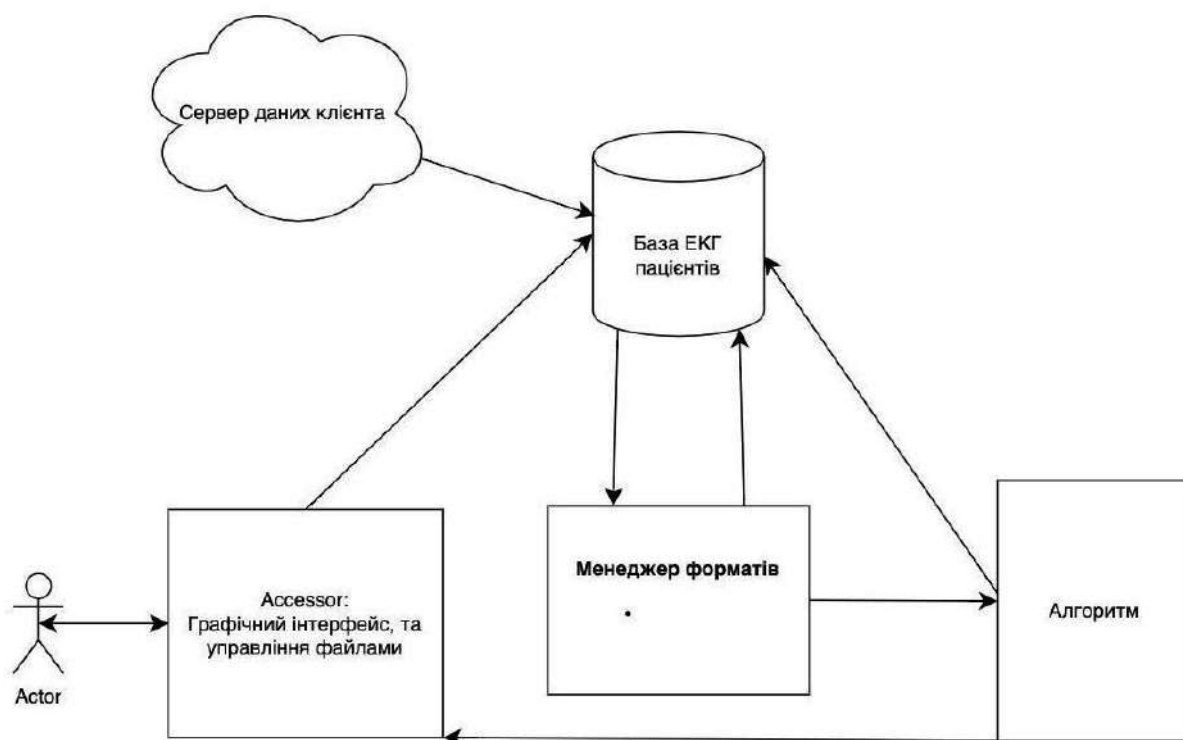


Рис. 4. Архітектура сервісів застосунку аналізу ЕКГ.

графічний інтерфейс, через який «актор» (лікар) завантажує файли з ЕКГ та отримує результати аналізу сигналу. Назвемо такий сервіс «Accesor», оскільки серед його задач є не лише відобразити, зчитати файл та занотувати проаналізований файл, а й упевнитись, що доступ до системи мають лише певні кваліфіковані люди. Такий доступ до системи можна організувати, як звичайним SSO провайдером, так і простими методами авторизації. Перед початком розробки GUI частини, що відображує сигнал було проаналізовано візуалізацію анотованих файлів бази МІТ-ВІН на ресурсі PhysioNet. Для зручності, сигнал розбивається на задані проміжки, переходячи між якими, можна роздивитись сигнал по частинах, наблизивши певні підозрілі ділянки.

На рисунку 5. зображено приклад, як виглядає графічний інтерфейс застосунку. Тепер, маючи представлення, з яких сервісів буде складатись застосунок, перейдемо до вибору технічних рішень, які можна використовувати для його реалізації. Для підтримки крос-платформленості було обрано реалізацію як веб застосунок. Таке рішення не лише уможливило збереження крос-платформленості, а й зберігає час потенційного користувача, якому не треба буде робити інсталяцію. Розгляда-

ючи тенденції мов програмування сьогодні, запропоноване рішення можна розробляти або використовуючи технологію nodeJS, або Flask/Django[13]. Було вирішено використовувати Flask, оскільки він написаний мовою Python, яка є популярною серед науковців, що займаються проблемами ЕКГ, та вже має готові бібліотеки роботи зі стандартизованими та поширеними форматами.

Відзначимо ще одну проблему, яку вирішує можливість авторизації.

У реальних умовах у великих лікарнях за день можуть приходити десятки ЕКГ для аналізу різних пацієнтів, прив'язаних до різних лікарів. Авторизація дозволяє відфільтрувати, які ЕКГ доступні користувачеві (в нашому випадку це лікар), що дозволить кожному лікарю бачити лише своїх пацієнтів та уникати плутанини.

Після проведення аналізу результатів будь-яких медичних вимірювань пацієнт зазвичай очікує опис від лікаря та звіт про зафіксовані патології або аномалії. Тому невід'ємною частиною застосунку, який проводить аналіз ЕКГ, є генерація звітності. Логічно цю задачу делегувати застосунку «Accesor», оскільки він комунікує з базою даних та отримує результати обробки ЕКГ алгоритмами аналізу.

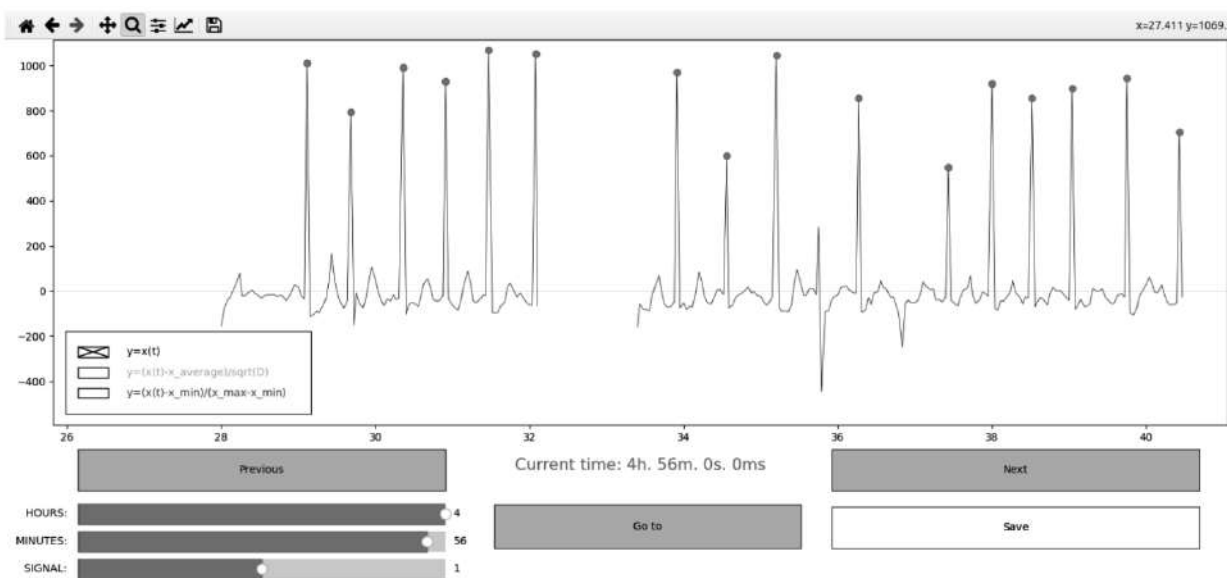


Рис. 5. Система візуалізації кардіограми з знайденими піками

Зазвичай звіт зручно генерувати або у PDF або .doc документи. Звісно, генерація таких документів є вже задачею складнішою за відображення, сортування файлів та внесення додаткової інформації, тому для цього варто створити окремий сервіс, який би з отриманих результатів обробки ЕКГ та додатково доданих лікарем анотацій та коментарів, генерував документи для пацієнтів. Підкреслимо, що декодовані ЕКГ є звичайними масивами з чисел, які можна зберігати у реляційній базі даних, то самі файли ЕКГ є складними бінарними структурами, тому треба використовувати сервіс зберігання даних. Це може бути як хмарне рішення, так і FTP сервер тощо. Оскільки декодуємо й отримуємо ЕКГ як масив чисел, з якого потім можна відтворити сигнал, то зберігати основні файли довгостроково немає сенсу. Такий підхід дозволяє використовувати сховища даних невеликих обсягів, які час від часу вичищаються. Поповнення таких сховищ можна реалізувати як через «Accesor», так і напряму з серверів даних лікарень, якщо у лікарні такі є. Беручи до уваги описані сервіси разом, – така мікро-сервісна архітектура дозволяє значно прискорити та автоматизувати роботу кардіологів, що приведе до покращення роботи з пацієнтами.

Висновки

Розроблено та запропоновано підхід для автоматизації роботи кардіологів у лікарнях шляхом розробки системи, яка дозволяє інтегрувати у зручному для лікарів форматі алгоритми аналізу ЕКГ у їх роботу опису кардіограми. Система дозволяє не лише здійснювати аналіз ЕКГ у пришвидшеному темпі, а й систематизувати файли різних форматів, які можуть бути наявні у лікарнях. Запропонований підхід дозволяє обробляти різні типи кардіограм та конвертувати їх у єдиний формат, що спрощує їхнє подальше використання алгоритмами роботи з ЕКГ. Розроблено крос-платформний застосунок, який дозволяє продивлятися фрагменти кардіограми, з детальною візуалізацією та алгоритмічно доданими нотатками, з мож-

ливістю додавання анотацій лікарями напряму, якщо алгоритм аналізу ЕКГ пропускає деякі ділянки. Система підтримує роботу із стандартизованими форматами ЕКГ та має можливість інтегрувати будь-який формат, якщо відома його специфікація. Архітектура побудована так, що дозволяє інтегруватись у готові рішення, які наявні у лікарнях, що дозволяє отримувати повну історію пацієнтів. Такий підхід забезпечує більш ефективне управління даними пацієнтів, що сприяє зниженню кількості помилок та підвищенню якості надання медичних послуг.

Даний сервіс був протестований на розробленому алгоритмі пошуку R-піків, для демонстрації його точності у порівнянні з іншими алгоритмами. Використання описаного сервісу не лише пришвидшує роботу кардіологів, а й систематизує зберігання ЕКГ у зручному вигляді та започатковує вирішення проблеми автоматизації аналізу медичних вимірювань, дозволяючи додавати алгоритми обробки та підтримку будь-яких форматів, не змінюючи основну архітектуру системи.

References

1. Goldberger, A.L. et al. “PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals.” *Circulation*. Vol. 101, 23 (2000): E215-20. doi:10.1161/01.cir.101.23.e215
2. Liu, F. et al. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 8(7), (2018): 1368-1373. doi:10.1166/jmihi.2018.2442
3. Cai, Z. et al. An Open-Access Long-Term Wearable ECG Database for Premature Ventricular Contractions and Supraventricular Premature Beat Detection, *Journal of Medical Imaging and Health Informatics*, 10(11), (2020): 2663–2667. DOI: 10.1166/jmihi.2020.32892663
4. Mu, H. et al. Electrocardiogram Minnesota codings from 30 000 adult cases with Kazakh ethnicity in Xinjiang, China. *Zhonghua liuxingbingxue zazhi*. 31(4), (2010): 451-454 PMID: 20513295.

5. ANSI/AAMI EC11:1991/(R) (2001) *Diagnostic Electrocardiographic Devices, 2ed*
6. Kemp, B., Olivan, J. European data format 'plus' (EDF+), an EDF alike standard format for the exchange of physiological data. *Clinical Neurophysiology*, 14(9),(2003):1755-1761. [https://doi.org/10.1016/S1388-2457\(03\)00123-8](https://doi.org/10.1016/S1388-2457(03)00123-8).
7. Xie, C. et al. Waveform Database Software Package (WFDB) for Python (version 4.1.0). (2023). *PhysioNet*. doi/10.13026/9njx-6322
8. Philips Healthcare SERVICE BULLETIN. SB86000222A: 860292, 860323, 989803190581 - *Release Holter 3.0.3*. 2017.
9. Health Insurance Portability and Accountability Act [HIPAA] of 1996, Pub. L. No. 104-191.
10. Consolidated text: Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).
11. Krak, Iu., Pashko A., Stelia O., Barmak, O., Pavlov, S. Selection Parameters in the ECG Signals for Analysis of QRS Complexes. *1st International Workshop on Intelligent Information Technologies and Systems of Information Security, InteIITSIS 2020*, Khmelnytskyi, 10-12 June 2020. Vol. 2623. 2020, pp.1-13. <https://ceur-ws.org/Vol-2623/paper1.pdf>
12. Pashko, A., Krak, Iu., Stelia, O., Khorozov, O. Isolation of informative features for the analysis of QRS complex in ECG signals. *Advances in Intelligent Systems and Computing*, Vol. 1246 AISC, 2021. pp. 409 – 422. DOI: 10.1007/978-3-030-54215-3_26.
13. Grinberg, M. (2018). *Flask web development: developing web applications with python. x27*; Reilly Media, Inc.

Одержано: 02.09.2024

Внутрішня рецензія отримана: 11.09.2024

Зовнішня рецензія отримана: 13.09.2024

Про авторів:

^{1,2}Єфремов Микола Сергійович,
аспірант, асистент кафедри
<https://orcid.org/0000-0001-8698-3957>

^{1,2}Крак Юрій Васильович,
член-кореспондент НАН України,
д.ф.-м. н., професор,
завідувач кафедри
<https://orcid.org/0000-0002-8043-0785>

Місце роботи авторів:

¹Київський національний університет імені
Тараса Шевченка,

²Інститут кібернетики імені В.М.Глушкова
НАН України

Тел.: +380 44 239-33-33

e-mail: Iurii.krak@knu.ua, Yefremov@knu.ua

К.І. Кузьміна, Т.М. Сьомик, А.П.Андон

РОЗРОБКА МОДУЛЯ СОЦІОПСИХОФІЗІОЛОГІЧНОЇ ПІДТРИМКИ ДЛЯ СУЧАСНИХ ІТ, ЯК ЗАСОБУ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ДІЯЛЬНОСТІ ОСОБИСТОСТІ І КОЛЕКТИВУ ПРИ ЗБЕРЕЖЕННІ ЇХ БІОСОЦІАЛЬНОГО ЗДОРОВ'Я

Основна ідея роботи – наукова організація праці та відпочинку сучасної людини, сім'ї, школи, колективів, світової спільноти. Така організація є перехідним (від середньостатистичного до індивідуально-типологічного) моментом на шляху ефективності організації тандему «Професійна компетентність – біосоціальне здоров'я людини». Мета роботи – розробка модуля СПФ підтримки для сучасних ІТ. Вона розглядається як засіб підвищення ефективності діяльності особистості і колективу при збереженні їх біосоціального здоров'я. СПФ модуль – технологія принципово нова і розглядається в контексті автоматизації управління. Досліджується можливість адаптації розробленої технології «Сімейна медицина на дому» до структури СПФ модуля як надійного сучасного способу СПФ підтримки біосоціального здоров'я трудового колективу. У ширшому масштабуванні технологія «Сімейна медицина на дому» розглядається, як початок принципово нового розуміння проектною культури сьогодення щодо медицини та охорони здоров'я людини.

В контексті цифрової трансформації інформаційного потоку представлені об'єктивні СПФ показники індивідуально-типологічного та ритмологічного портрета колективу. Серед них кореляційні та частотні плеяди, резонансна частота і швидкодія системи та інше. СПФ модуль – технологія повинна забезпечити формування реального комп'ютерного міждисциплінарного СПФ комплексу. Це сприятиме підтвердженню пріоритетів у сучасному людинознавстві таких характеристик, як індивідуальність особистості, біоритмологія та комп'ютеризація, що може сприяти суттєвому підвищенню економічного та культурного потенціалу держав.

Ключові слова: мікро- та макро-агресивне середовище, тандем «професійна компетентність-біосоціальна культура», динамічна структура особистості, персонолог-міждисциплінарник, адаптивний моніторинг, СПФ протокол здоров'я, сімейна медицина, СПФ модуль, квантова V-образна крива.

Kuzmina K.I., Somik T.M., Andon A.P.

DEVELOPMENT OF A MODULE OF SOCIOPSYCHOPHYSIOLOGICAL SUPPORT FOR MODERN IT AS A MEANS OF INCREASING THE EFFICIENCY OF INDIVIDUAL AND COLLECTIVE ACTIVITIES WHILE PRESERVING THEIR BIOSOCIAL HEALTH

The main idea of the work is the scientific organization of work and leisure of a modern person, family, school, teams, and the world community with a transitional (from average to individual-typological) moment on the way to the effectiveness of the organization of the tandem «Professional competence - human biosocial health». The purpose of the work is to develop a SPF support module for modern IT as a means of increasing the effectiveness of the activities of an individual and a team while preserving their biosocial health. SPF module is a fundamentally new technology that is approaching automation of management. The possibility of adapting the developed technology «Family medicine at home» in the structure of the SPF module as a reliable modern highly competent SPF support method (interdisciplinary) for preserving the biosocial health of the labor collective is being considered. In a broader scale, the technology «Family Medicine at Home» is considered as the beginning of a fundamentally new understanding of the project culture of today, regarding medicine and human health.

In the aspect of digital transformation of the information flow, objective SPF indicators of the individual-typological and rhythmological portrait of the collective are presented: correlation and frequency constellations, resonant frequency and system speed, etc. SPF module - technology should end with a real computer interdisciplinary SPF complex to confirm the priorities in modern human sciences such as human individuality, biorhythmology and computerization, cybernetics, which will contribute to a significant increase in the economic and cultural potential of states.

Keywords: micro- and macro-aggressive environment, tandem «professional competence-biosocial culture», dynamic personality structure, interdisciplinary personologist, adaptive monitoring, SPF health protocol, family medicine, SPF module, quantum V-shaped curve.

Вступ

Основна ідея роботи – наукова організація праці та відпочинку сучасної людини, сім'ї, колективів різного рівня, світової спільноти. Така організація є прогресивним, високораціональним моментом сьогодення, засобом підвищення ефективності тандему «Професійна компетентність – біосоціальна культура людини (як високоінноваційний спосіб підтримки першої)». Удосконалення професійного протоколу – вищих та індивідуальних властивостей людини – це по суті «високоінтелектуальне обличчя» кожного сервісного міждисциплінарного комп'ютерного продукту, що забезпечує високий життєвий комфорт сім'ї та обіцяє суттєве підвищення економічного та культурного потенціалу не лише України, а й усієї світової спільноти.

Мета роботи — розробити науково-практичне обґрунтування та перспективи комп'ютерної реалізації здоров'язберігаючої технології «Сімейна медицина на дому» на основі універсального соціопсихологічного (СПФ) модуля як початку принципово нового етапу проектної культури охорони здоров'я людини. **З позицій індивідуально-типологічного підходу**, принципів цілісності, системності, динамічності, ритмологічності та міждисциплінарності, універсальності СПФ модуля розглянути професіограму, динамічний СПФ портрет сім'ї, її індивідуальний адаптаційний інформаційний простір (ІАП) – раціональне для життєдіяльності середовище (дизайн) на основі нової проектної культури.

Сучасний соціум перебуває в стадії інтенсивних змін та пошуку перспективної стратегії розвитку нових технологій та соціальних перетворень, що є вкрай пріоритетними для інформаційного суспільства.

Ефективність інформаційного суспільства як соціальної системи, заснованої на знаннях, передбачає високий рівень організації (системогенезу), взаєморозуміння та узгодженої гармонійної взаємодії, взаємодії її компонентів.

Інформатизація переростає в соціально-технологічну революцію, сам результат якої переміщується у фокус філософських досліджень, розуміння сутності людського буття, соціопсихологічної адаптації і дезадаптації, ментального здоров'я людини, вдосконалення її трудових умінь. Свідомість стає активною стороною універсуму, що самоорганізується. Видатний вчений Глушков В. М. задовго до ери інформаційного суспільства передбачав бурхливий розвиток ІТ та їх входження до всіх сфер сучасного життя.

Необхідна нова освіта, нові підходи до здійснення навчально-виховного та трудового процесу, формування інноваційного нелінійного мислення, корисності, плідності й цікавості.

У даній роботі відображений аспект: «методологія – технологія – техніка» для ефективної організації логічно, системно обґрунтованого виховання, освіти професіонала і соціопсихологічної підтримки його професійної компетентності протягом всього життєвого циклу з урахуванням міждисциплінарності.

Згідно із сказаним, частка організаційної системного середовища та реалізація ідеї розглядаються на основі еволюційних та інноваційних знань сьогодення. В їхній основі - розвиток людини як особистості (основний критерій суспільного прогресу); глобалізм з позицій СПФ основ людини; здоров'язберігаючі технології та комп'ютерні системи в сучасному людиноведенні для раціональної організації життєвого циклу людини та колективу; цифрова трансформація інформаційного потоку, знань про головний мозок, як основну управляючу ланку організму.

1. Розвиток людини, як особистості – основний критерій суспільного прогресу

Сучасна філософія освітян – визначальний фактор авторитету соціальної держави. Найпріоритетнішими у 21 столітті

стають наука як сфера, що продукує нові знання, і освіта, що олюднює знання і, на-самперед, забезпечує індивідуальний розвиток людини.

Таким важливим компонентом планети як особистість розглядається спосіб буття справжнього гуманізму, основа утвердження свободи людини як соціально-активної, соціально-відповідальної. **Духовність** – це чітко визначена орієнтація внутрішнього свята особистості на практичне здійснення основного імперативу гуманізму: для людини це ціль, самоціль. Інтелігентність – практичне здійснення, або гуманізм в дії, сплав інтелекту та практичної дії. Сьогодні світ дуже складний, біфуркаційна картина з очевидними загрозами в економічній і соціальній сферах, істотна зміна в науковій карті світу, погляд на світ як на складне ціле; затвердження нелінійного мислення, що розвивається.

Різні знання соціопсихології, персонології, інформатики, біоніки, синергетики, біоритмології, еволюційної спадковості готові вже зараз стати новою основою подальшого розвитку сучасної людини та наук про неї (біофілософія, біополітика, біоісторія, біоекономіка тощо).

Принципово новий напрямок знань – типологія особистості – типологія середовищного дизайну – інноваційний напрямок у людиноведенні. Світ давно вже уявляється не сумою об'єктів, а сумою відносин. Ефективність колективного результату визначається мистецтвом компромісу протягом усього його життєвого циклу.

Одна з основних граней у структурі творчої особистості – воля до дії, зміни, розвитку. **«Щоб світ постійно поставав перед людиною як світ завдань і проблем, а вона була б здатна, спроможна їх виконувати... треба спиратись цілком на наукове теоретичне пізнання і нести момент естетичної насолоди, естетичної завершеності»** [1].

Розглядаються еволюційні механізми процесів, подібних у всіх системах, забезпечених єдиними фундаментальними законами. Знання спільності законів організації формує принципово новий вектор розвитку людини та людства. Цивілізація

неухильно рухається до побудови нового суспільства (від homo erectus, homo sapiens до homo intelligence), де вирішальну роль відіграють здібності та якості особистості, що утверджуються в світі.

Проектувати світ, індивідуальний адаптивний інформаційний простір (ІАІП), виходячи з типології особистості, законмірностей (загальних і часткових), її генно-фенотипологічної організації протягом усього життєвого циклу – принципово новий напрямок у людинознавстві та людиноведенні.

Науково-освітня сфера має стати престижною, здобуття нових знань – ключовий фактор для функціонування держави та її стратегічної безпеки. Інтелект має стати суспільним пріоритетом. Необхідно нове формування світу: наука у цьому процесі особливо пріоритетна. На думку вчених «політикам необхідно це зрозуміти та підтримувати науку щедро та розумно».

2. Глобалізм з позицій соціопсихологічних (СПФ) досліджень людини

Глобалізм розглядають як процес стирання географічних кордонів, формування соціального та культурного світогляду, поширення однотипних моделей розвитку на всі країни планети, дифузії технологій, введення мономови.

Відомі вчені вважають, що поняття «глобалізм» було б не таким розпливчастим, якби філософи, історики та політологи звернулися до міждисциплінарних знань адаптації людини.

Виходячи з того, що матерія, єдина у своїй організації, можна використовувати структуру біосистем для пояснення організації соціосистем (за аналогією). Прихильники глобалізму вірять у глобальне громадянство, у те, що проблеми людства можна вирішити демократичним глобалізмом. Це ідея про те, що всі люди мають право, незалежно від того, де живуть. А загальна свобода та права людини можуть сприяти усьому людству.

«Образ глобалізації» постає як фрагмент континууму функціональних станів міжпланетарного симбіозу цивілізацій. В

історії людства і на основі наших знань – найбільш напружена біфуркаційна стадія еволюції інформаційного потоку.

Все, характерне для окремого індивідуума, може бути віднесено і до нації: сила і слабкість нервової системи; лідерство та свідомість, емпатійність, стеничність, емоційна стійкість, рівень контролю, реалізаторський, творчий потенціал, особливості біосоціального здоров'я.

Виходячи з того, що особистість є структурно-функціональною одиницею нації, знання про тип особистості та її динамічну структуру формує принципово новий антропоекономічний, антрополітичний, антропоісторичний погляд на управління суспільством, виходячи з його гено-фенотипових властивостей.

Відомі СПФ портрети *прибалтійських та нордичних національностей*, що відрізняються гіпоемоційністю та ірраціональністю. *Американська популяція* більш раціональна й емоційна, практична, не схильна до ідеалізації, в міру товариська, формальна в контактах, помірно емпатійна (співпереживання). *Російська* - гіперемоційна, значною мірою фрустрована, схильна до ідеалізації, у спілкуванні виявляє полярні тенденції: від любові до ненависті, тобто надзвичайно пристрасна [2]. *Українці* соціоніками визначені як етико-інтуїтивний інтроверт із його замкнутістю, серйозністю, скромністю, емпатією, емоційністю, талантом, відданістю. Вдосконалення української етнопсихології має йти шляхом більшої самостійності, раціональності, ініціативності, відкритості, аналітичного мислення, посилення масової «хліборобської» складової національного характеру цінностями «лицарської» складової [3].

Вища влада, що управляє, повинна вирізнятися лідерством, високою інтелегентністю і інтелектуальністю, вимогливістю, екстравертністю, вмінням захищати і неодмінно захочувати.

Кібернетики називають тип – режимом управління. Вибір структури соціального управління, формації передбачає групування на СПФ знаннях про типологію нації, її гено-фенотипологічну обумовленість, а режим управління – згідно з нею, а не всу-

переч. Людина чи нація, лише тоді реалізується у житті, якщо сенс її життя збігається з її природним покликанням та СПФ можливостями. Тоді вона талановита, унікальна, адекватна, тоді вона здорова, ефективна та щаслива. Водночас напрошується аналогія з напруженістю фізіологічних процесів: жорстко детермінований режим функціонування суспільства без ступеню свободи, але із заохоченнями, більш адекватний до витривалих, з жорсткою структурою внутрішньої системної організації (параноїдальних осіб).

Схожий (конвульсивний) стан ми спостерігаємо зараз на планеті за так званого стану глобалізму: висока геомагнітна напруженість планети, залучення до цієї напруженості всіх цивілізацій в економічному, політичному та соціальному планах; це катастрофи (повені, землетруси), зниження народжуваності, суттєве посилення поляризації (бідний - багатий).

Продовжуючи розглядати такий режим (глобалізації) з позиції СПФ механізмів адаптації людини, можна бачити, що цей режим дуже високого ступеня напруги та взаємозв'язку між компонентами системи (високими показниками фазових запізнювань за експериментальними даними) свідчить про критичний рівень деструктивного стресу (дистресу), хаосу (по Пригожину), вираженої дисгармонії, функціональної декортикації, зокрема «мозку» (або зниження можливостей керуючої влади), можливого провісника значного порушення саморегуляції будь-якого організму (людина, сім'я, нація, планетарний симбіоз). Це режим так званого «холостого ходу», коли вихід системи слабо результативний, або зовсім його немає через низьку енергетичність забезпечення адаптації та фрагментарного мислення.

Матерія вічна, кінця її не передбачається, але катаклізми «судомна активність, постконвульсивна депресія» очевидні. Тут працює об'єктивний ритмологічний закон і адаптація у широкому сенсі виражається V-подібною перевернутою кривою, відбиваючи активацію, напругу, перенапругу, втому, формуючи скрізь і всюди циклічні процеси [4,5].

Це стик проблеми адаптації та проблеми біоритмів. Остання безпосередньо пов'язана з типологією особистості та відповідно – нації. Є і люди, і нації із пластичними та інертними властивостями нервової системи, гнучкою та жорсткою структурою внутрісистемної організації, а єдиний ендогенний синтетичний внутрісистемний ритм (фінальне злиття приватних ритмів різних систем) є своєрідним індикатором типу особистості, нації, їхнім морфофункціональним відображенням.

Біоритмологічний типологічний адаптаційний портрет нації буде орієнтиром її біосоціального здоров'я, економічного потенціалу, адекватності управління та симбіозу планетарних цивілізацій.

Для адекватної синхронізації між країнами в усіх галузях (але не на рівні високої глобальної синхронізації) дуже важливий ступінь напруги систем, взаємин між їх елементами. Це завжди має бути активація (але не перенапруга).

Головною атрибутикою нової архітектури світової спільноти має бути не уніполярна держава, не децентралізований світ, оскільки надмірний порядок і жорсткість або повний хаос та анархія – це структури нестійкі, приречені на вмирання. Середнє між ними – це оптимальна структура СПФ життєздатності світової спільноти. Достатньо адекватним такий режим видавався років 12–15 тому. За аналогією із СПФ механізмами йшлося про доволі високий рівень комунікації між країнами з високою енергетичністю, творчим потенціалом, адекватним рівнем надходження та переробки інформації. Режим оптимальної працездатності системи, якщо це синхронність і синфазність, то управляючі впливи стають дещо неформальними (а управління - рентабельним). Ось тут можна було б звернути увагу на ступінь демократії, щоб вона не перейшла на вседозволеність і не сприяла надмірній глобалізації. Можливо, цей аспект біосоціальної культури людини тоді був базовим і процес перебудови був більш лабільним.

А взагалі цей режим – оптимальної працездатності світової спільноти досить енергетичний (але не надмірно, як за гло-

балізації) є перехідним, біфуркаційним. Це мобілізаційний режим, максималізація зусиль для створення фону, на якому буде реалізовуватись основний кінцевий результат системи: як при адаптації людини: *у разі умовного рефлексу – його реалізація, тренування – закріплення навичку, харчової поведінки – насичення, у операторській діяльності – вирішення завдання, втома – перехід на найнижчий функціональний рівень* [4,5].

Особливості адаптаційної взаємодії керівників і виконавчих ланок (держави та народу), визначаючи загалом оптимальну структуру симбіозу, також безпосередньо пов'язані з їхньою типологією. Якщо вгорі ієрархічної піраміди правитель із низьким рівнем синергії (співуправління), то управління зосереджується в руках неадекватних людей.

Насправді ритмічність вносить принципово нове в багато сфер теоретичного і практичного розуміння знань про матерію. Якщо раніше поняття «життя» - це (за Енгельсом) спосіб існування білкових тіл, сьогодні – це спосіб існування коливальних процесів. Наші СПФ можливості – це відображення їх (коливальних процесів) взаємодії та взаємосприйняття. З точки зору ритмології все СПФ забезпечення життєвого циклу будь-якої системи відображено у стадіях різнорівневої V-подібної кривої. На ній можна побачити зростання напруги системи в період її оптимальної працездатності, її граничну пропускну здатність (біфуркаційний момент) з подальшим спадом напруги та депресивними тенденціями. Біоритмологія як системотворчий фактор наближає науку до цифрової трансформації інформаційного потоку, чіткіше відповідає на питання оптимізації діяльності людини, напрямів її захворювань, запобігання падінню тривалості життя [6].

Людина сьогодення та завтрашнього дня має прагнути статусу homo intelligence - людини інтелігентної з високим рівнем СПФ культури, творчим та реалізаційним потенціалом, посиленням к.п.д. *індивідуального та колективного розуму*.

В основу останнього мають бути покладені такі високорангові людські якості, як емпатія (почуття співпереживання),

взаємодія, гнучкість мислення, тактичне та стратегічне мислення, інформативний інноваційний потенціал.

Система планування та управління, спрямовані на гармонізацію людини, колективу, здійснюється фактично за залишковим принципом (5% держбюджету), тоді як у високорозвинених країнах на ті ж цілі йде ~ 20%. Дефіцит «формування» людини (всебічне виховання, гармонізація особистості, зниження рівня СПФ культури) веде до спотворення світорозуміння, формування бездуховності та жорстокості, девальвації вічних ідеалів.

3. Здоров'язберігаючі технології і комп'ютерні системи в сучасному людиноведенні для раціональної організації життєвого циклу людини

Сьогодні, в століття комп'ютерної цивілізації проблема персонології стоїть надзвичайно гостро. Психологи вважають, що саме існування людської цивілізації залежатиме від *біосоціальної грамотності людства, його розуміння самого себе і навколишнього середовища*. Тому найважливішим завданням сучасних персонологів є проблема людського капіталу, **біосоціальної культури людини** - знання про себе, свою індивідуально-типологічну адаптацію в навколишньому світі, що дозволяє щасливо, з меншою психофізіологічною вартістю прожити своє життя, реалізувати свої можливості [4,5].

Водночас перспективне міждисциплінарне вивчення поведінкових реакцій людини і їх забезпечення: гено-, фено-, фізіо-, біохімічне та ін. [3,4], що активно відходить від середньостатистичного підходу [5].

Важливі знання про динамічну структуру особистості і її індивідуальний адаптаційний інформаційний простір (ІАП) як про перспективну проєктну культуру «типологія особистості - типологія середового дизайну» [4,5]. Суть останньої в смисловій, колористичній і композиційній реалізації комфортного індивідуально-типологічного простору людини [8,10].

Час підготував об'єктивний ґрунт для систематизації, уніфікації різноаспектних біосоціальних знань про людину в теоретичному плані. А **міждисциплінарний лонгітюдний соціопсихофізіологічний комп'ютерний експеримент**, заснований на гено- фенотипологічній платформі, внесе принципово нове в розуміння цілісності людини, раціональне і ефективне використання цієї платформи в усі періоди її життєвого циклу [4,5,9,10].

Тобто основна ідея роботи повинна окреслити інноваційні перспективи в теоретико-емпіричному дослідженні **особистості, її цілісності, системності, динамічності, хвилеподібності, міждисциплінарності** та широкому використанні цих знань для створення високоінтелектуальних інформаційних здоров'язберігаючих технологій та комп'ютерних систем [5,9,10].

3.1 Про структуру особистості в сучасній персонології

Із численних варіантів визначення особистості провідними психологами сьогодні найбільше імпонують такі [11]: Г. Олпорт (1937) «*Особистість* - це внутрішня динамічна організація психофізіологічних систем, яка формує патерни поведінки, мислення і емоцій, характерні для даної людини». Олпорт дає визначення особистості як *зрілої людини* з реалістичним сприйняттям, завершенням розпочатої справи, самопізнанням, почуттям гумору. За Келлі (когнітивний напрям) люди - думаючі істоти, людина по суті - вчений, дослідник, інтерпретатор доквілля[11].

Обговорення питань, пов'язаних із розумінням структури людини, спрямоване до знань *її рис і типу* [12]. В наш час розглядаються 2 пласти знань: теоретико-емпіричні про основні положення щодо природи людини [12] (рис. 1) і теоретико-експериментальні про нейрофізіологічні механізми, об'єктивні показники умовно-рефлекторної діяльності (школа Павлова, Теплова, Небиліцина, Анохіна та ін. [13,14]).

Ці знання, спрямованість теорій укорінені в особистому індивідуальному досвіді їхніх творців. Кожна риса людини представлена континуумом положень (або

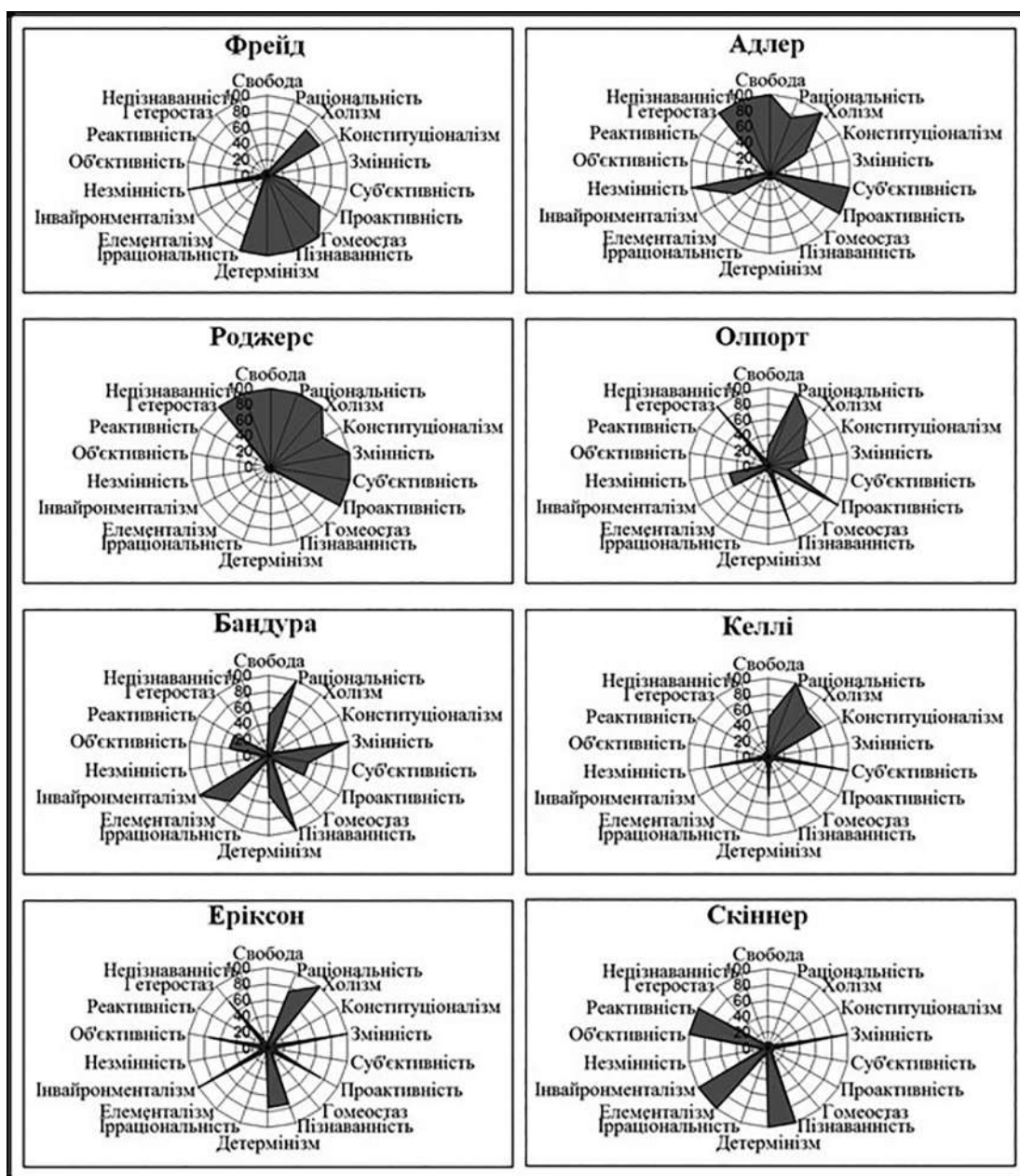


Рис. 1. Огляд позицій провідних теоретиків з дев'яти основних положень, що стосуються природи людини (80 - 100 - сильна, 60 - 80 - помірна, 40 - 60 - середня, 20 - 40 - слабка)

властивостей особистості) з двома екстремумами (наприклад, «свобода» розташована на одному полюсі першого континууму, а «детермінізм» - на іншому полюсі цього ж континууму, тип особистості може визначатися 2, 3, 6, 8, 16 індивідуальними властивостями (Айзенк - 3, ММРІ - 13, Юнг - 8, Кеттел - 16, Павлов - 3, 12).

Інша спрямованість робіт щодо структури особистості, зокрема, вивчення темпераменту, веде нас до припущень Гіппократа (V ст до н.е.) про відмінності лю-

дей за співвідношенням чотирьох основних «соків» організму - крові, флегми, жовтої та чорної жовчі з виділенням типів: *холерик*, *меланхолік*, *сангвінік* і *флегматик* (Клавдій Гален, II в н.е.). Подальші дослідження у царині диференціальної психофізіології отримали відображення в працях [13,14]. В основу індивідуальних відмінностей покладені фундаментальні властивості нервових процесів збудження і гальмування (сила, їхня врівноваженість і рухливість).

Небиліцин В.Д. [14] у своїй концепції, акцентуючи увагу на динамічній стороні питання, вказує на те, що комбінації процесів збудження і гальмування в нервовій системі складають нейрофізіологічну основу «всіляких психологічних проявів».

Виділено [13,14] 4 основних типи вищої нервової діяльності: *сангвінік* - сильний, врівноважений, рухливий тип нервової системи; *холерик* - сильний, нерівноважений, рухливий; *флегматик* - сильний, урівноважений, інертний; *меланхолік* - слабкий, нерівноважений, інертний.

Є припущення, що сильній нервовій системі відповідає симпатична, а слабкій - парасимпатична реактивність [15]; показано зв'язок врівноваженості нервових процесів з особливостями вегетативного реагування.

Розглядають загальні властивості: активність, емоційність, емоційна стійкість [16]. Поняття «активність», «активування» і «активація» визначають енергетичний рівень роботи організму, мозку і організацію нервових процесів. Активність пов'язана з рівнем коркового збудження, з такими поняттями як інтро- і екстраверсія.

Показано [16], що у *інтровертів* вище рівень коркової активності, і ця ознака є стійкою індивідуальною властивістю; *екстраверти* більш активовані у разі вираженої монотонії і високого інтересу до діяльності. Рівень активації у екстра- та інтровертів є внутрішньою причиною їхньої поведінки (пасивність-активність).

Поняття «оптимум збудження» [4,5] або «оптимум стимуляції» - це рівень збудження, оптимальний для життєдіяльності індивіда з найменшими фізіологічними витратами. Індивід підтримує цей рівень різними способами. Показано, що високий рівень збудження має своїм біохімічним субстратом високу концентрацію норадреналіну і дофаміну, високу концентрацію ендорфінів, які знижують больові відчуття.

Емоційність особливо гостро відчувається під час стресу, недостатній продуктивності під час напруженої діяльності, «афектах неадекватності» і відповідних їм форм поведінки - ригідності, розгубленості, впертості, негативізму, дратівливості,

загальмованості тощо, що зумовило виникнення проблеми емоційної стійкості [16].

Айзенк [17] вважає, що нейротизм, екстраверсія, інтроверсія обумовлені генетично, а крайні їх форми ведуть до патології. Посилення нейротизму й інтроверсії веде до дистимії, що проявляється в нав'язливості і страхах. Високий рівень нейротизму на тлі високої екстравертності супроводжується ознаками психопатії та істерії. Гіперзбудливість (нервозність) проявляється у формі руховій, пізнавальній і емоційній.

Більш глибокі уявлення про психофізіологічну структуру особистості розглянуті в роботах Л.Н. Собчик зі співавторами [18]. Так надмірність показників стеничної саморегуляції загрожує дезадаптацією за *соціально-поведінковим типом*, протилежне співвідношення означає відмову від самореалізації і переважання соціального тиску на особу, формує *невротичний патерн дезадаптації*; у разі перенапруження тих чи інших механізмів відбувається їх взаємне «гасіння» і формується *біологічний варіант дезадаптації* з психосоматичними розладами.

Одним із способів вивчення біологічного і психічного є використання патологічного матеріалу [19] зі збереженням принципу: від закономірностей здорової психіки до патології. Автор виділяє кілька типів: *шизофреніки* - відсутність спрямованості на пошуки правильної інтерпретації, є тільки формальна констатація елементів; *параноїки* - схильність до деталізації, надмірного «моралізування», до гіпертрофованих оціночних суджень; *психопати* - рівень домагань дуже швидко формується, завищений, але відрізняється крихкістю; *істеріки* - конфлікт між рівнем домагань і неможливістю його реалізації, слабка цілеспрямованість, ефективність, розпорошеність, різка неузгодженість у колективі, відсутність волевого зусилля, недостатня самокритичність, неадекватність поведінки.

Розробки уявлень про *біосоціальну культуру* людини і її *основ*, включаючи знання про тип особистості, її динамічну структуру й індивідуальний адаптаційний інформаційний простір (ІАП) в різних сферах сучасного людинознавства - це наступ-

ний, реальний для реалізації пласт знань щодо вдосконалення та гармонізації сучасної людини [4,5,9,10].

3.2 Функціональний стан організму

Проблема функціональних станів стара, як світ, і водночас актуальна. Еволюція визначення цього поняття зазнала змін від простого - це «гло, на якому розвиваються поведінкові акти людини і тварин», чи «реалізується конкретна діяльність» до складнішого - «інтегральний комплекс готівкових характеристик тих функцій та якостей, які прямо чи опосередковано зумовлюють виконання трудових процесів» [4,5].

Основу уявлень про континуум функціональних станів створюють дані, отримані в рамках теорії активації. Запропонована гіпотетична шкала рівнів «неспанья-сон-перезбудження» поєднує величезний діапазон поведінкових реакцій у зв'язку з рівнем активації різних систем організму, причому активацію пов'язують з рівнем нервового та соматичного функціонування, ступенем енергетичної мобілізації [4,5]. Зниженню працездатності поряд із втомою сприяють *монотонія та психічне пересичення*.

Сельє [Цит. 4,5] розглядав стрес як «неспецифічну реакцію організму на будь-яку вимогу ззовні». Біоритмологічним індикатором стресу надзвичайно високого рівня є збільшення амплітуди ритмічних процесів за рахунок інтенсифікації внутрішньоклітинного метаболізму та залучення до діяльності резервних функціональних структур, спочатку з неповною, а потім – з тотальною взаємною синхронізацією [4,5]. Масований викид гормонів кори надниркових залоз у разі психічної травми так сильно порушують катаболізм білків, що токсичні речовини, що утворюються (аміак, азот залишковий тощо), є основою депресивних і маніакальних станів, шизофренічних синдромів. В активації стресової реакції велику роль відіграють мигдалеподібний комплекс, який підвищує адренкортикальну активність гіпофіза, в ослабленні стресу – зниження активності ретикулярної формації та медіальної ділянки гіпоталамуса. Переважання у більшості тварин парасимпатичних впливів

в умовах емоційного стресу запобігає аномалії серцевої діяльності (захисна дія блукаючих нервів на серці оберігає від артеріальної гіпертензії). Позитивна роль стресу (еустрес) - у мобілізації ресурсів «захисту» життєво важливих систем організму.

Одним із найбільш поширених методів досліджень функціональних станів головного мозку є електроенцефалографія, а її частотно-амплітудні характеристики – загальноновизнаними їхніми критеріями.

Слід зазначити великий внесок М.М. Ліванова та його школи у розшифровку феномену просторової синхронізації та її різних проявів [20]. Вперше 1966 року у його лабораторії було здійснено так званий керований експеримент, у результаті було доведено функціональне значення коефіцієнта кореляції. М.М. Ліванов сформулював правило відповідності функціональних станів. Для *оптимальної передачі* в корі головного мозку необхідна відповідність рівнів лабільності, фазових відносин та високого ступеня синхронізації біопотенціалів взаємодіючих зон і, у разі досліджень на кролику, наявність кореляції в смузі тета-ритму.

Розуміючи механізми організації як структури особистості ЛО, так і його відповідної діяльності, надзвичайно важливо розглядати *синхронізацію як загальнобіологічну закономірність*, що лежить в основі часової самоорганізації процесів живої системи (рис. 2) [4,5]. І хоча синхронізація не стала ще предметом цілеспрямованого теоретичного вивчення, біоритмологія нині постає як новий підхід до дослідження складноорганізованих біопроцесів, що вимагає розробки теорії **біологічної синхронізації**.

На шляху створення такої теорії розглядаються системні концепції, універсальні способи взаємодії між осциляторами - резонансні взаємодії. А самому явищу синхронізації належить «видатна організуюча роль» [20]. Важливі не власне коливання, а їхня взаємодія, що приводить до їхньої самоорганізації. Енергетичні характеристики систем досягають своїх екстремальних значень у стані синхронізації. У разі збільшення об'єму системи коливання з швидкозгасаючих перетворюються на повільно-

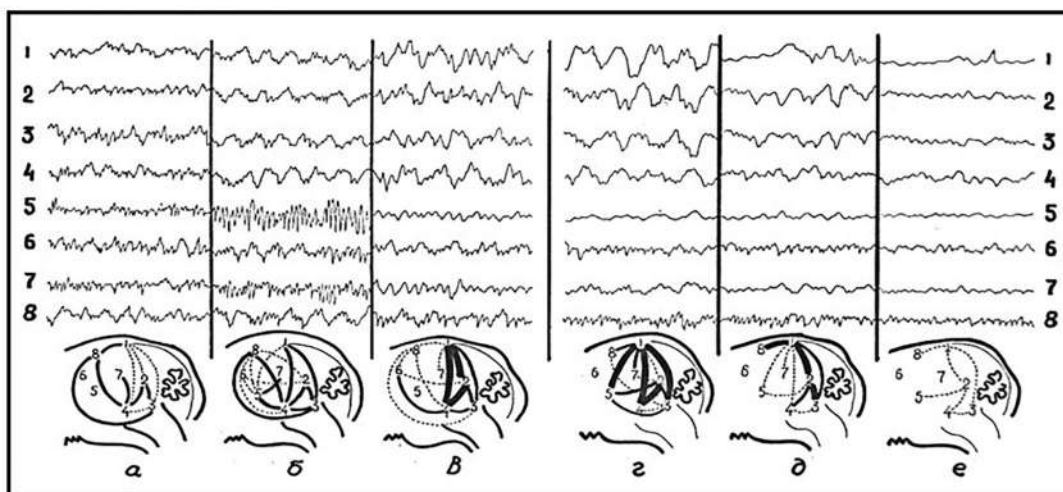


Рис. 2 Внутрішньомозкові кореляції фонового стану /а/, в різні стадії (I-V) нембуталового наркозу (б, в, г, д, е) за показниками ЕЕГ. На схемі мозку зображені всі значущі зв'язки: слабкі /0.2-0.3/ - пунктир; середні /0.31-0.49/ - тонкі лінії; сильні /0.5-1/ - жирні лінії; 1 - NC мот., 2 - R, 3 - NRT, 4 - SGC, 5 - APL, 6 - NSM, 7 - P, 8 - Hip.d

згасаючі, а далі - на незгасаючі. Можливо це механізм судомної готовності та судомної активності [8,10]. Зазначено, що синхронізація ритмів сприяє полегшенню іррадіації процесів і збудження, і гальмування. Режим дистантної синхронізації однотипно проявляється у процесі формування адаптивних станів організму: при виробленні умовних рефлексів, розумової діяльності, фізичного навантаження, природному (ортодоксальному сні), харчовій та статевій поведінці, під час дії наркотичних речовин [10].

А.Б. Коган [21] пов'язує такий режим (глобальної синхронізації, коли мозок працює як єдине ціле) з об'єднанням великих аналізаторних систем ділянок кори в ритмі «холостого ходу», коли структури гіперсинхронізовані, напружені, а вихід - безрезультатний. Подальше значне зниження активності коркових утворень (децентралізація) переводить систему на режим судомної активності з наступним можливим відновленням. Показано [4,5], що у режимі генерації епілептиформної активності відбувається заміна негативного зворотного зв'язку на позитивний, в основі чого лежать патологічні зміни у системі саморегуляції. І, нарешті, з урахуванням розглянутих заяв про механізми синхронізації як *системоутворюючої функції* організму можна констатувати наявність індивідуально-типологічної *біоритмічної кривої працездатності* ЛО, а біоритм

виступає як інструмент підтримки якісної стійкості живої системи, феномена *хвилеподібності* адаптаційного процесу [20, 21].

3.3 Людина та її біосоціальна культура

Біосоціальна культура людини – це насамперед знання про себе, свої долереалізуючі тенденції, можливості, резерви, стійкість до стресу, вміння адекватно формувати міжособистісні стосунки та багато іншого. По суті, це знання свого міждисциплінарного соціопсихофізіологічного портрету.

Біосоціальна культура людини несе в собі потужний арсенал засобів самопізнання, самоаналізу, самоврядування для успішної адаптації людини в навколишньому середовищі, істотного зниження її психофізіологічної вартості.

Новий еволюційний етап пізнання людини має базуватися на індивідуальному соціопсихофізіологічному підході, знаннях про структуру особистості та континуум її функціональних станів, відображених у понятті «*динамічна структура особистості*» [4,5].

Відповідно до концепції «типологія - континуум функціональних станів» необхідно вивчати і властивості, і стани, які ними забезпечуються. Кожна типологічна структура (наприклад, *холерик, флегматик, сангвінік і меланхолік*) повинна пройти свій «*адаптаційний коридор*» для

отримання індивідуальних кількісних характеристик діяльності її різних психофізіологічних систем у випадку різних впливів, що наблизить до розуміння механізмів індивідуальної адаптації та суттєво підвищить адекватність керування функціональним станом людини.

Практичний аспект може бути представлений *соціопсихофізіологічним портретом особистості*, який робить істотний внесок у її біосоціальну культуру. Соціопсихофізіологічний портрет особистості формується за допомогою автоматизованої системи багатостороннього дослідження особистості (АСМІЛ), створеної в Інституті програмних систем НАНУ та дає вичерпну інформацію про особистісні властивості та актуальний стан обстежуваного [5,9,10].

Водночас важливими є й інші аспекти індивідуального підходу до здоров'я [4,5,9]. Це, по-перше, вивчення здоров'я та хвороби як *єдиного динамічного процесу*; по-друге, під час медичної корекції є можливість оцінити актуальний стан, характер реакції хво-

рого на хворобу, обрати спосіб спілкування з ним і метод лікування, контролювати адекватність корекції функціонального стану [5,9,10].

Виходячи з того, що особистість є структурно-функціональною одиницею нації, знання про типологію особистості та її динамічну структуру формує принципово новий **антропосоціальний погляд на управління суспільством**, виходячи з її генотипових властивостей [3,4].

Людина з огляду на свою індивідуальну психофізіологічну організацію відображає світ по-різному і розряджається різними каналами — вісцеральним, руховим та інтелектуальним. Тому енергію індивіда необхідно спрямовувати у потрібне, соціально прийнятне русло. Таким руслом, найголовнішим керуючим фактором, джерелом життєстійкості людини та нації є **праця**. Наукова організація праці на основі знань про генотипові можливості людини обіцяє і високу ефективність діяльності, і збереження її біосоціального здоров'я [5,9,10].

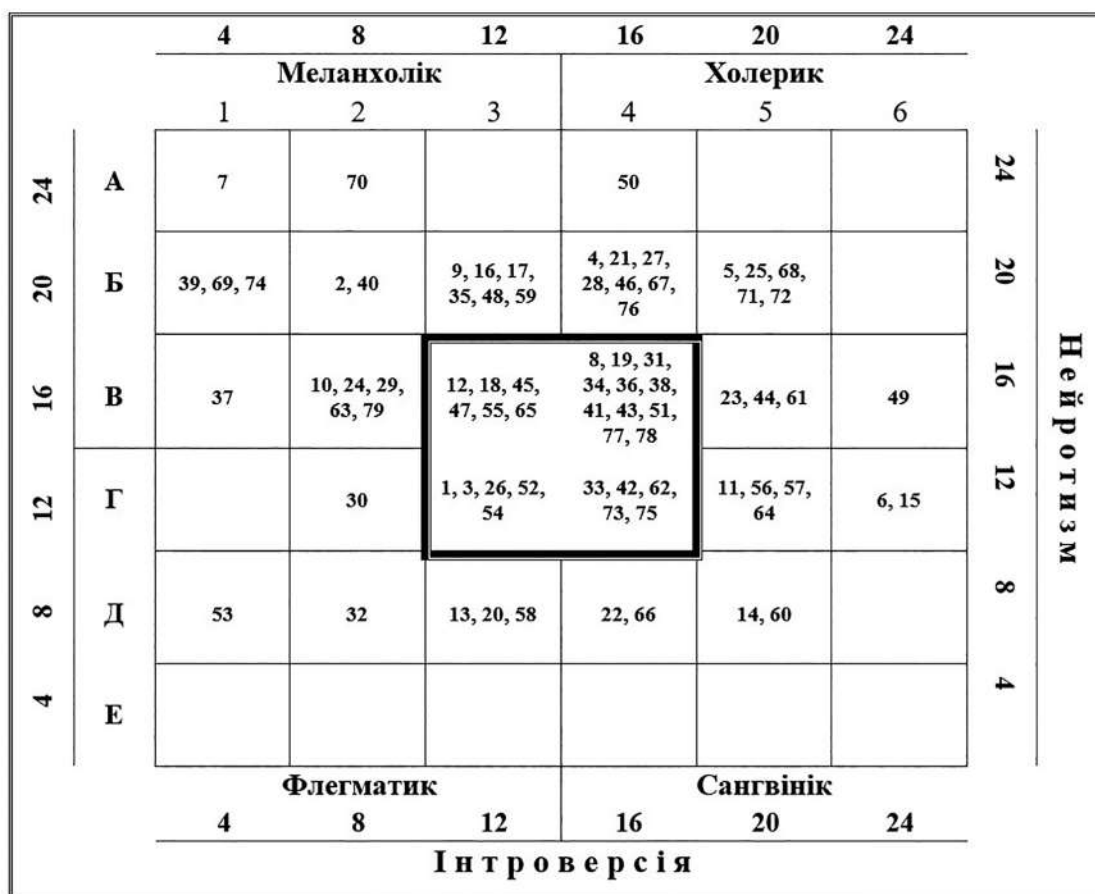


Рис.3.а Загальна картина розподілу типів особистості студентів-медиків на схемі Айзенка

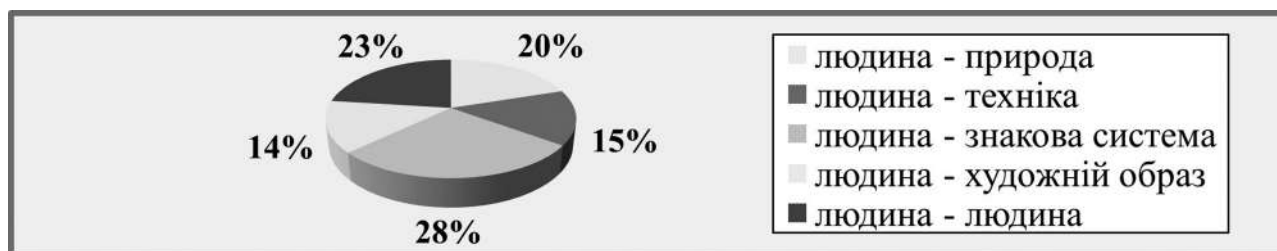


Рис.3.б Показники профорієнтації студентів (по 6 групам)

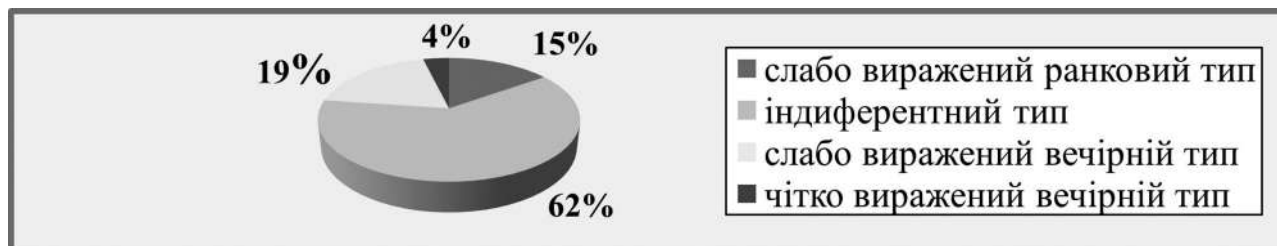


Рис.3.в Активометричні характеристики студентів (по 6 групам)

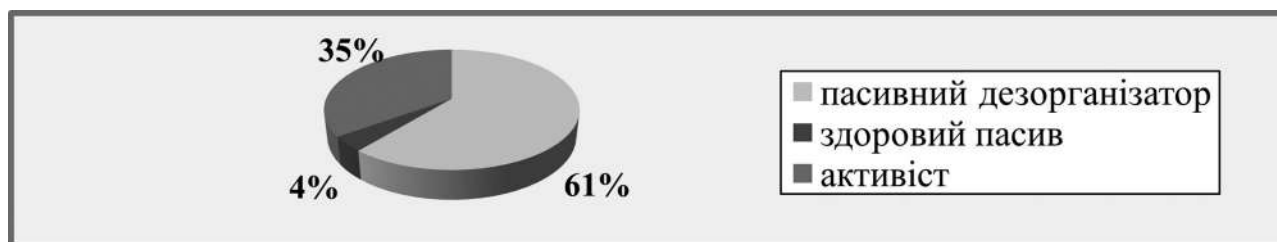


Рис. 3.г Загальна картина розподілу показників «сови-жайворонки» (по 6 групам)

Індивідуально-типологічний підхід до пізнання людини має розглядатися в руслі державної програми охорони здоров'я та особливо здоров'я здорової людини та орієнтувати на підвищення рівня біосоціальної культури нації, її морального та економічного потенціалу.

Міждисциплінарність – різнорівневе забезпечення процесу адаптації людини (гено-, фізіо-, соціо-, нейро-, біохімічне та інших) - новий напрямок – універсальна наука, про який давно мріяли вчені, здатна об'єднати у собі основні наукові дисципліни [5,9,10].

Прогресивним інструментом для реалізації принципово нових можливостей розуміння механізмів соціопсихологічної адаптації людини, оцінки її функціональної надійності, зокрема, у людино-машинних (ЛМ) системах, є *комп'ютерні технології* [9,10, 23].

Соціопсихологічні (СПФ) дослідження авторів технологічно забезпе-

чуються комп'ютерними системами АСНІ, ЕФЕКТ, КОЛІР, КОМФОРТ, які за допомогою системи Конструктор можуть бути використані в різних послідовностях, діапазонах та поєднаннях пред'явлень. На рисунках 3, 4, 5 представлені приклади результатів комп'ютерного аналізу [9,10].

На підставі цього для *інтроверта* (рис.5) буде адекватним та комфортним гомогенне візуальне середовище з домінуванням ахроматичних кольорів у колористичній палітрі: ненасичені (пастельні) кольори малюнків та фону, слабка інтенсивність відео- та аудіо супроводу. Дуже важлива гармонійність композиції, поділ текстової частини та декоративного фону, естетичне та смислове рішення. Велика різноманітність елементів (рамки, лінійки, кольорові планки) та їхня висока динамічність виключаються.

Для *екстраверта* (рис. 5) – все навпаки: різноманітний ілюстративний матеріал, динамічність агресивного (актив-

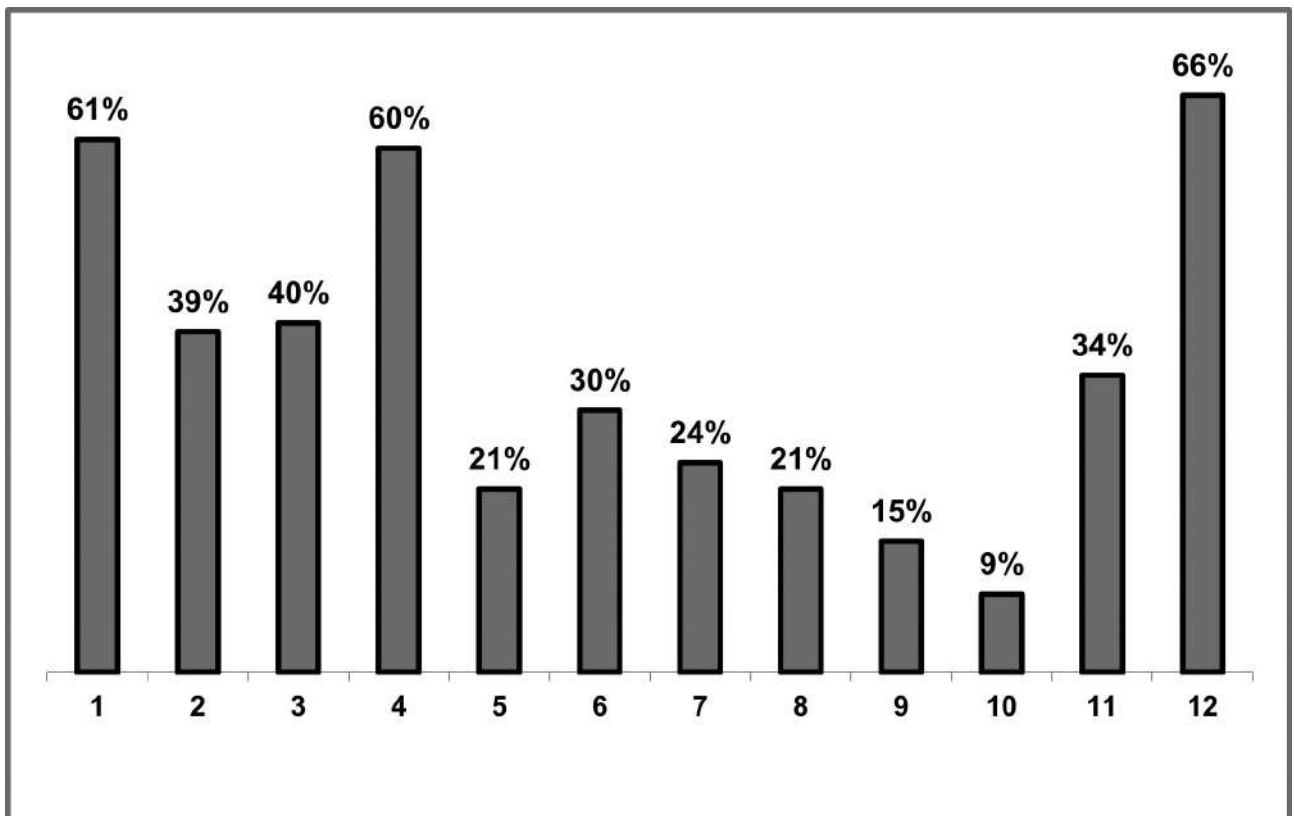


Рис. 4 Деякі соціопсихофізіологічні особливості колективу (у відсотках від загальної чисельності колективу):

- 1 – витривала нервова система; 2 – слабка і середня витривалість нервової системи;
 3 – «провідні»; 4 – «ведені»; 5 – явні лідери; 6 – творчо окреслені індивідуальності;
 7 – «мужні» жінки; 8 – нормальний жіночий стиль статевої поведінки;
 9 – «жіночі» чоловіки; 10 – «дуже мужні» чоловіки; 11 – інтроверти; 12 – екстраверти.

ного) середовища, достатня дисгармонія, еkleктичний (анархічний) дизайн, головне – яскравість та динамічність пред'явлення інформації, мажорний аудіосупровід. Такий *індивідуальний інформаційний простір* оптимізує, рятує від монотонії, підвищує рівень активації мозку екстраверта, що суб'єктивно сприймається ним як комфорт.

Для профілактики таких фрагментів функціонального стану як пересичення, звикання та монотонія (особливо для користувача із сильною нервовою системою) необхідно згодом, не змінюючи основного змісту, вводити стеничні (активні) компоненти.

3.4 Висновки

Здоров'язберігаючі технології та комп'ютерні системи – основна ідея представлено широкого аналізу знань із теоретичних та прикладних проблем раціональної організації життя людини протягом її життєвого циклу.

Аналізуючи скаже, не перестаєш дивуватися генію великих вчених у всі віки, їхній високій інтуїції та творчій компетентності.

Так, ще в V ст до н.е. Гіппократ говорив про відмінності людей за співвідношенням чотирьох основних «соків» організму - крові, флегми, жовтої та чорної жовчі з подальшим виділенням типів: *холерик, меланхолік, сангвінік і флегматик* (Клавдій Гален, II ст. н.е.). Далі приходить розуміння загального характеру – універсальності принципу розвитку; погляд на світ як єдине ціле, яке складно розвивається, підпорядковується єдиним закономірностям розвитку. Цікаві ідеї Фрейда у тому, що живі організми є динамічними енергетичними системами, які підпорядковуються законам фізичного всесвіту; за Адлером кожна людина має певний енергетичний рівень, у межах якого вона здійснює наступ на свої життєві

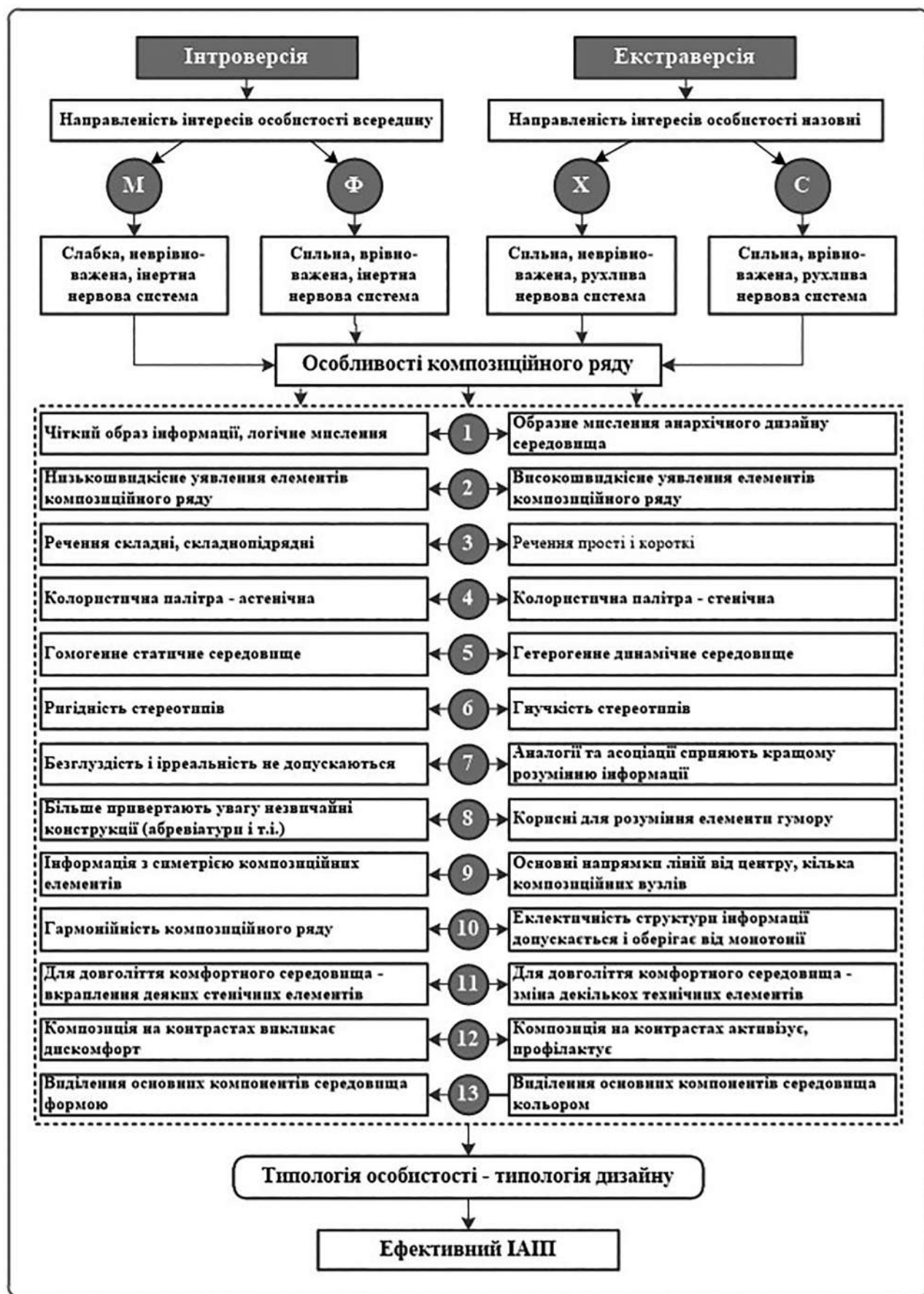


Рис. 5 Індивідуально-типологічні особливості композиційного ряду ІАП для інтровертів та екстравертів (М – меланхолік, Ф – флегматик, Х – холерик, С – сангвінік)

проблеми; в теорії Адлера соціальний інтерес виступає основним критерієм психологічної зрілості людини. Періодизація хімічних елементів Д.І. Менделєєва, яка відображає універсальний закон природознавства, може впорядковувати знання про системи організму та передбачати їхню дезадаптацію. Ймовірно, періодизація відбиває ступінь зрілості тієї чи іншої науки.

За П. Капіцею, відомим вченим нашого часу, «... Розвиток науки про закони, які є на чолі організації суспільства, має ґрунтуватися на науці про *вищу нервову діяльність*». Тільки на цій науковій основі можна створити колективи для правильного виховання і навчання людей. Тільки на цій науковій базі можна шукати правильні форми організації праці і відпочинку людини. І головне, **тільки на науковій основі можна створити здорову, ефективну структуру суспільства**».

Персонологія дуже молода галузь знання, продукт ХХ століття, вже досягла повноліття, заявивши про себе як життєздатна і велика сфера досліджень. Уявлення про неї як про «наречену без приданого», «вершника без голови», «яка не вийшла з періоду дитинства», істотно послабилася, хоча 1981 року відомий вчений В. Налімов вважав, що «Незнання людини може бути найбільш сильним незнанням сучасної науки». Розуміння людської поведінки, міжособистісних стосунків в групах різного рівня сьогодні стало неодмінною умовою для виживання нашого роду.

Сказане переконує, що сьогодні біфуркаційний, перехідний, «проривний» період не тільки в еволюційній глобалізаційній схемі планети, а й в періоді розуміння індивідуальності людини, в тому числі її ролі в «новонароджених» науках (біосоціології, біополітики, біоекономіки, біоісторії та ін.).

Наразі дозрів фундамент для цілісного системного міждисциплінарного розуміння людини на підставі **індивідуально-типологічного** (а не середньостатистичного) підходу. Він повинен продовжувати свій шлях від homo erectus, homo sapiens до homo intelligence - людини, з такими високоранговими людськими якостями, як

емпатія (співпереживання), як мистецтво компромісу і взаємодія, стратегічне і тактичне мислення, раціональність, висока лабільність, соціальний, творчий, реалізаційний потенціал. Духовність і інтелігентність (постійне вдосконалення інтелекту і віддача його результатів людям) розглядаються як вектори успішних гуманістичних і практичних реалізацій. **Біосоціальна культура** - знання про себе і свої індивідуально-типологічні адаптаційні можливості мають стати **високоранговою компетенцією сучасної людини ХХІ століття**.

Серйозні зміни, що відбуваються сьогодні у всіх сферах життя, підвищення освітнього і культурного рівня людства, демократизація і комп'ютеризація, зокрема, вимагають звернення до потенційних, науково-обґрунтованих можливостей людського капіталу, який розкривається з резервами гуманізації, інтелектуалізації, підвищення економічного потенціалу суспільства, гармонізації біосоціального здоров'я.

Уніфікація і систематизація знань із питання, що вивчається, привела до розуміння того, що з'являється універсальна *цілісно-орієнтована сучасна наукова технологія (методологія, методика та програма)* вивчення індивідуально-типологічної адаптації людини та колективу на основі знань про її цілісність, ритмічність, міждисциплінарність, системність, динамічність, і з широкою практичною комп'ютеризацією, яку ми назвали **«Соціо-психологічний СПФ модуль»** [9,10,23].

Є надія, що достатня зрілість міждисциплінарних знань про людину, динамічну структуру її особистості, індивідуальний адаптаційний інформаційний простір (ІАП), готовність сучасних технологій та комп'ютерних систем зорієнтовують дослідників і, особливо, керуючих ланок країн на принципово нове розуміння, адекватні програми з людинознавства і людиноведення. Зокрема, створення науково-практичних центрів біосоціальної культури людини, сімейної медицини, професійної компетентності для вищих ешелонів управління, дружніх інтерфейсів, реклами, центрів гармонізації колективів різних рангів та багато іншого.

Сьогодні найбільш актуальна проблема персонологів - гармонізація людського капіталу, симбіозу його високої професійної ефективності і біосоціального здоров'я. Вивчення механізму цього симбіозу, його активна комп'ютеризація (здоров'язбеігаючі технології і системи) стануть проривним, високораціональним моментом на шляху наукової організації повнокровного, щасливого способу життя людини.

4. На шляху цифрової трансформації інформаційного потоку

4.1 СПФ підтримка підприємства та колективу

В аспекті цифрової трансформації даних та знань «Індустрія 4.0» навіть не торкається людського фактору, вдосконалення якого сьогодні вже за допомогою нашої комп'ютерної СПФ технології може призвести до суттєвого підвищення рівня продуктивності праці та збереження біосоціального здоров'я колективу [4,9,10,22,23] (рис. 6).

Ми випередили час, ще в 70-90 роках минулого століття ми пропонували тех-

нологію вивчення протекторного ефекту електронаркозу при найгострішій променевої хворобі (1966), функціональної організації головного мозку під час переходу від неспання до наркотичного стану на базі імплантованих у мозок електродів (1978). Досить ефективним було дослідження зорового аналізатора з позиції ТАУ для диференціації наркотичних речовин - зворотний зв'язок, резонансна частота, швидкодія мозку тощо (1977).

Сьогодні під час реалізації систем у промисловому та цивільному будівництві на кожному етапі проектування, зведення об'єктів, монтажу, складання, моніторингу ходу робіт та технічних прийомів можливо залучати високопідготовлених за СПФ технологіями фахівців. Водночас виникає потреба у створенні методів ведення широкомасштабного комплексного моніторингу, діагностики та управління всією сукупністю технології Індустрії 4.0 для забезпечення належної якості та надійності функціонування підприємства загалом – це досконала інноваційна тема управління цілісністю цифрового підприємства [9,10].

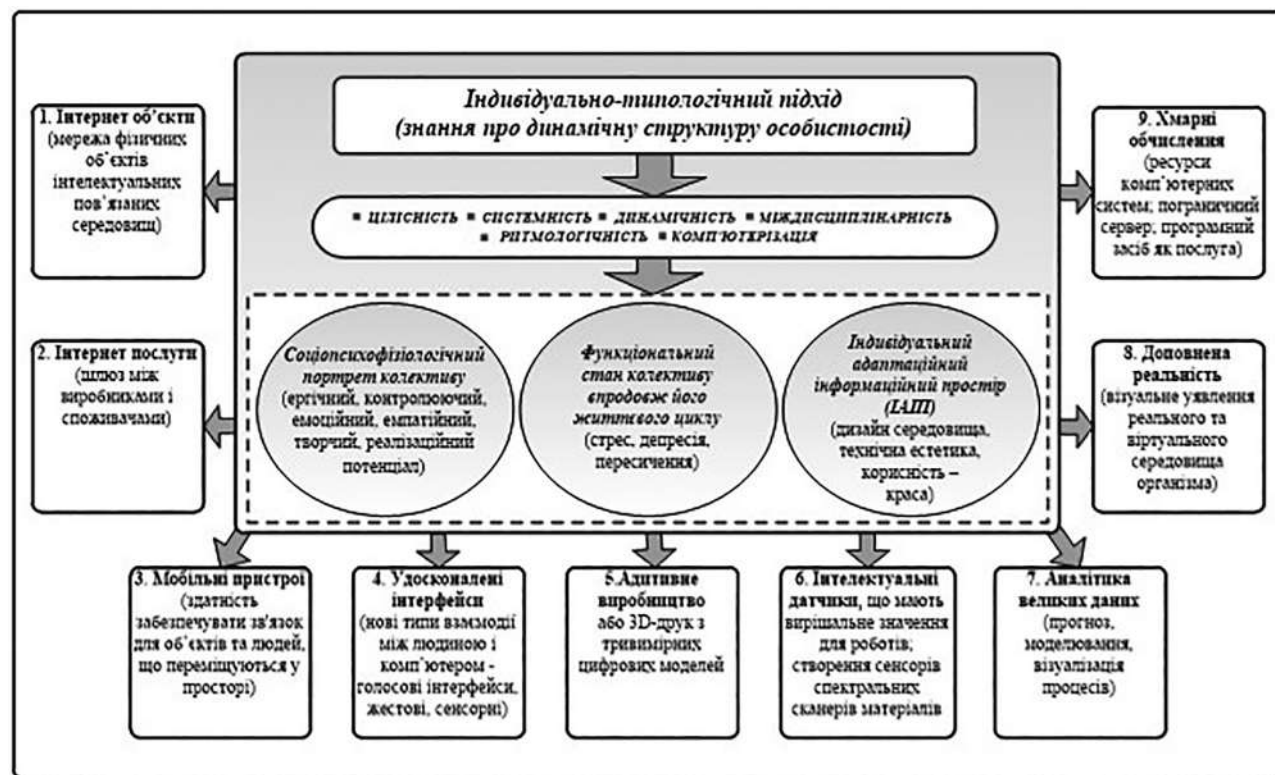


Рис. 6 Організація життя людини на соціопсихологічних знаннях

Дія сучасних систем цифрових технологій підприємства та колективу має відбуватися в масштабі реального часу процесів, які вони забезпечують. Якісна зміна, що привносить цифровими технологіями в життєзабезпечення країни, це високий ступінь інтеграції областей менеджменту, взаємне проникнення та злиття виробничих та цифрових технологій. Це створює умови формування єдиного інтегрованого, інформаційно пов'язаного, детально і точно представленого в усіх аспектах колективу, що динамічно змінюється в реальному масштабі часу (цифровий потік колективу).

4.2 Системотворча синхронізація в живих системах

Розуміючи механізми організації як структури особистості, так і її відповідної діяльності, надзвичайно важливо розглядати синхронізацію як загальнобіологічну системотворчу закономірність, що лежить в основі часової самоорганізації процесів живої системи.

І хоча синхронізація ще не стала предметом ціленаправленого теоретичного вивчення, **біоритмологія** зараз виступає як новий підхід до дослідження складноорганізованих біопроектів, міждисциплінарної концепції – теорії біологічної синхронізації.

Принцип екстремальних силових полів В.В.Белецького, інтегральний критерій стійкості Н.І.Блехмана, принцип найменшої взаємодії М.В.Овендена і класичний критерій стаціонарності А.Пуанкаре відображають одну спільну закономірність: під час синхронізації силові взаємодії досягають максимальних або мінімальних значень.

Імовірно максимуму відповідає фізичний стійкий стан, а мінімуму – нестійкий із ще довгим збереженням резонансних стосунків.

Вивчаючи механізми нейрофізіологічної і психофізіологічної адаптації людини й тварин за екстремальних впливів виявляємо надзвичайну подібність процесів синхронізації й саморегуляції із вище викладеними процесами [25]. Мабуть, знання цих дослідників про динаміку силових полів, принципи їхньої організації, знайдуть

відображення в індивідуально-типологічному динамічному цифровому портреті колективу і дадуть відповідь на його біосоціально-здоров'я. Можливо, це зорієнтує «людиноведів» на адекватні фізіотерапевтичні методи управління функціональним станом колективу.

У світлі розглянутих питань велике значення мають поняття «десинхронізація» і «синхронізація» (особливо дистантна) у формуванні різних функціональних станів головного мозку й системних реакцій організму [11,21].

5. Головний мозок – основна управляюча ланка організму

В розділі розглядається перспективність індивідуально-типологічного підходу для систематизації наявних, широких міждисциплінарних знань про людину та формування нових уявлень про норму, довголіття, прогноз особистості (її оптимальної працездатності, пандемії, спрямованості дезадаптації, адекватності відображення інформації, прогноз нових шляхів штучного інтелекту, здоров'язберігаючих технологій за рахунок індивідуалізації спілкування).

Сьогодні, в час здобуття глибоких теоретико-практичних знань про людину дозрів для розгляду феномен індивідуально-типологічна адаптація в біологічних, нейродинамічних, кібернетичних критеріях.

Пізнання людини стає актуальним щодо ембріональних уявлень під загальною назвою «Цифрова трансформація» наук. Наші 50-річні дослідження в усіх аспектах цього перспективного напрямку відображено практично у 300 друкованих роботах, 3-х дисертаціях, генералізованій науково-практичній апробації матеріалу в різних аспектах життєзабезпечення людини.

Досі акцент наших досліджень був спрямований на створення високонаукової здоров'язберігаючої технології, зокрема, на методолого-методичний аспект, а далі наші зусилля спрямовуються на управління функціональним станом людини, використання кібернетичних методів (зокрема, математичне моделювання).

Як методологічний результат - загальні стратегії індивідуально-типологічної

адаптації знаходять відображення в єдиному алгоритмі реагування, початкової активації центральних і вегетативних систем організму, наступній напрузі, стресі (стан найвищої напруги), перенапруги, відновлення або виснаження; особистісні стратегії визначаються різного ступеня виразності особистісних та часових показників (динамічних структурно-функціональних маркерів станів).

Особливу увагу приділено нейрофізіологічному забезпеченню різних стадій адаптаційного процесу [4,5,10], ролі ритмічних процесів, що відображають мобілізацію ресурсів організму для ефективного відновлення порушеного гомеостазу.

Мобілізаційний режим – це фрагмент континууму функціональних станів, який можна вважати стадією конструктивного стресу, режимом граничної пропускнуої спроможності оператора, що сприяє саморегуляції організму.

Подальша дія екзогенного фактору може призвести до деструктивного стресу – стану надмірної напруги та перенапруги з великими енергетичними витратами, вираженою дисгармонією систем, функціональною декортикацією головного мозку, зниженням кортикофугальних впливів кори на субкортикальні структури.

Останнє визначається наявністю універсальної загальнобіологічної закономірності або законом активності функціональних структур, що перемешуються. Внаслідок цього кожна окремо взята структура може періодично відновлювати свої матеріальні ресурси, виключаючи тотальну дистрофію.

Завдяки коливальному характеру фізіологічних функцій системи здатні «зондувати» різні якості системи, її властивості та стан зовнішнього середовища, безперервно адаптуватися до умов, що змінюються [26,27,28].

Сьогодні досить міцним горішком залишаються такі поняття як довголіття, адекватність відображення СПФ інформації, клімат Землі.

Якщо проаналізувати «адаптацію» дослідників (у тому числі й результати авторів проекту про індивідуально-типологіч-

ну адаптацію), то це буде V – образна крива, що відображає «сходи уявлення», ступінь активації та генералізації цих знань. В основі будуть праці Ухтомського, Введенського про нерв і клітину, на вершині велика кількість міждисциплінарних знань, цілих блоків і навіть конструкцій.

Запропоновано енергетичну теорію адаптивного реагування клітини та цілісного організму на стресори; концепція математичної моделі, що пояснює інтеграцію морфофункціональних систем організму для мінімізації дефіциту енергії [30].

Другим базисом, вельми методологічно інноваційним у рішенні проблеми енергобалансу нейровегетативних систем, є знання про індивідуальну адаптацію людини, її міждисциплінарний динамічний соціопсихофізіологічний портрет [4,5,9,10].

Очевидно для побудови **індивідуального адаптивного ритмологічного енергетичного** (метаболічного) портрета людини будуть отримані знання про **темперамент особистості**, що відображають генетично обумовлені **швидкісні і потужнісні властивості енергообміну організму**; знання про самоорганізацію організму (синергетику), синхронізацію як системотворчу функцію, що розкриває механізми еволюційних процесів, подібних у всіх системах та забезпечених **єдиними фундаментальними законами** [4,5,9,10,30,31].

Особливо корисне для розгляду завдань авторів знання про **типологічні групи і очікувані хвороби**, механізми біологічного годинника індивіда, типологію міжпівкульної асиметрії [31].

Цей матеріал особливо корисний для фахівців нейро- і біокібернетиків-«модельєрів» системної діяльності різних рангів використання кібернетичних методів (зокрема, математичного моделювання), озброєних комп'ютерною технікою. Одна з проблем зводиться до вивчення (за аналогією) взаємовідносин націй з урахуванням їхньої типологічної обумовленості та обліку глобального актуального ритмологічного процесу (екосередовище), поліметричної оцінки функціонального стану планетарних цивілізацій (у глобальному плані), прогно-

зування міжпланетарних конфліктів, пандемій та їх профілактики.

По суті ритмічність привносить принципово нове в багато сфер теоретичного і практичного розуміння знань про матерію. Якщо раніше поняття «життя» - це (за Енгельсом) спосіб існування білкових тіл, сьогодні – це спосіб існування коливальних процесів. Наші СПФ можливості – це відображення їхніх коливальних процесів, взаємосприйняття та взаємодії. З точки зору ритмології СПФ забезпечення життєвого циклу будь-якої системної реакції відображено у стадіях V-подібної перевернутої кривої.

На ній можна побачити зростання напруги, швидкодію системи, резонансну частоту, період оптимальної працездатності, граничну пропускну здатність (біфуркаційний момент з подальшим спадом напруги та депресивними тенденціями). Біоритмологія як системотворчий фактор наближає науку до цифрової трансформації інформаційного потоку, більш чітко відпо-

відає на питання оптимізації діяльності людини, напрямів її захворювань, запобігання падінню тривалості життя.

6. Основний лейтмотив СПФ підтримки технології «Сімейна медицина на дому»

Сформульована структура інформаційно-технологічної бази, що розкриває специфічні особливості сучасної інноваційної комп'ютеризованої медицини та охорони здоров'я (рис. 7,8,9).

Водночас головними принципами сімейної медицини є тривалість та безперервність спостереження за здоров'ям людини; багатопрофільність – ставлення до сім'ї як одиниці медичного обслуговування.

Людина розглядається, як цілісний образ, неподільне ціле у плані адаптації до різних факторів зовнішнього середовища; превентивність – основа діяльності сімейного лікаря, його профілактична спрямованість; прогнозування ризику розвитку окремих захворювань, їхня клінічна діагностика.

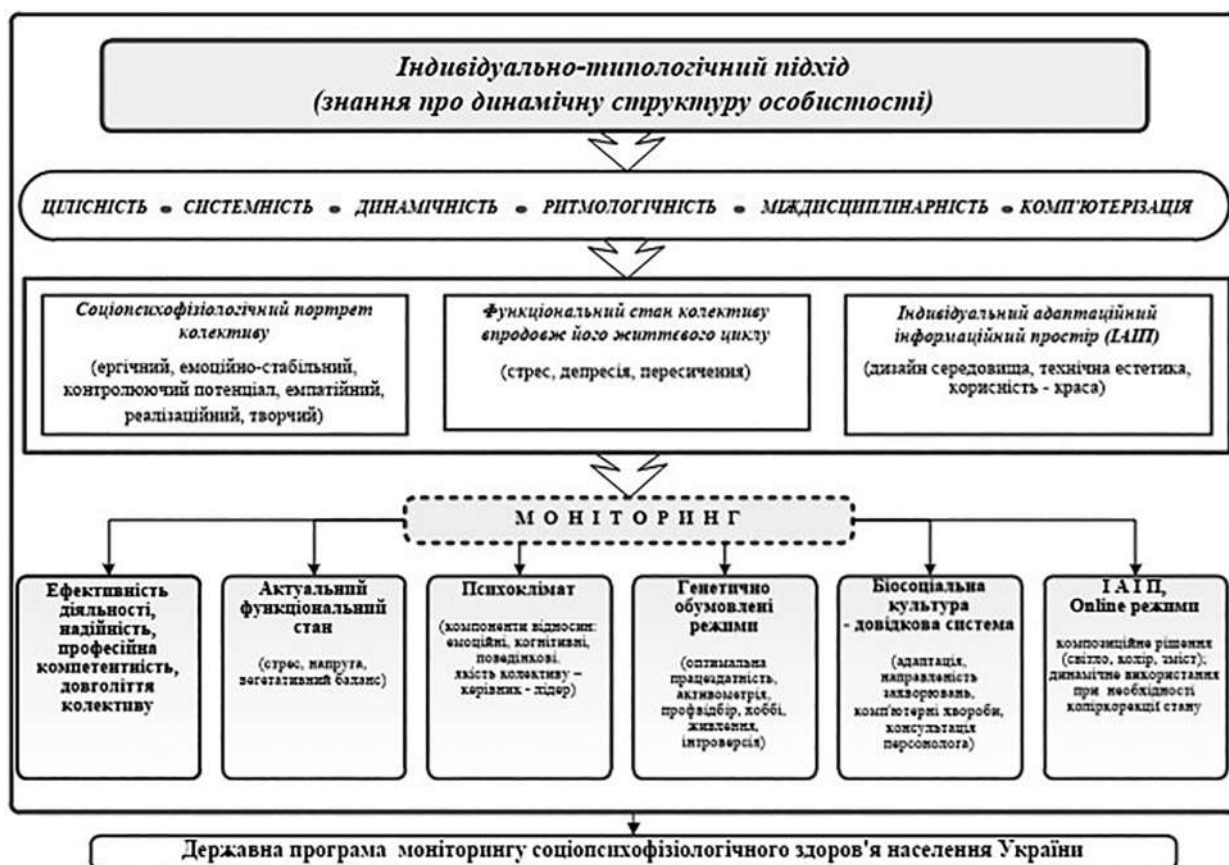


Рис. 7 Розроблення теоретичних основ для сучасної комп'ютеризованої організації трудового колективу (на основі СПФ модуля)

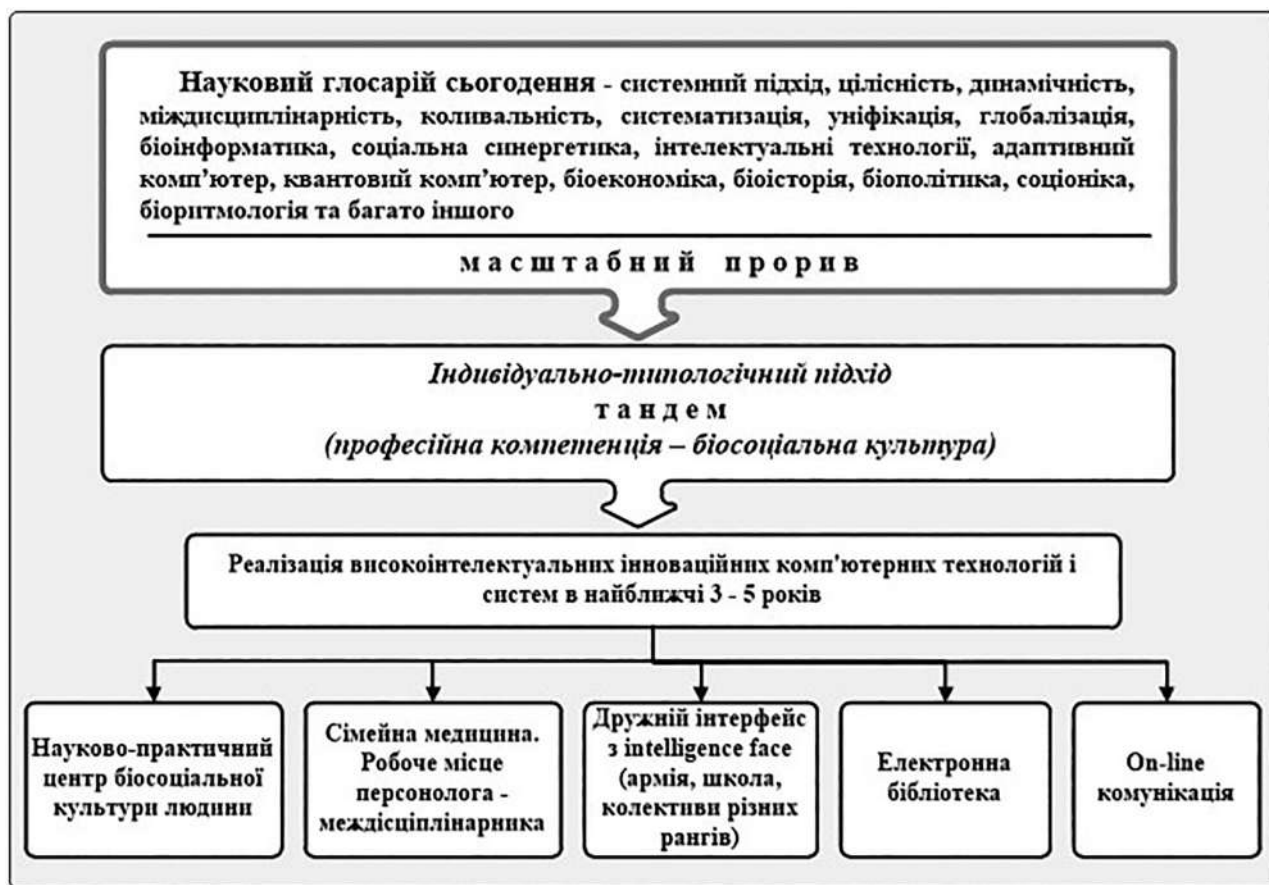


Рис. 8 Перспективність розробки універсальної цілісно-орієнтованої сучасної інформаційної технології (методологічна, методична та програмна) оптимізації людино-комп'ютерної взаємодії

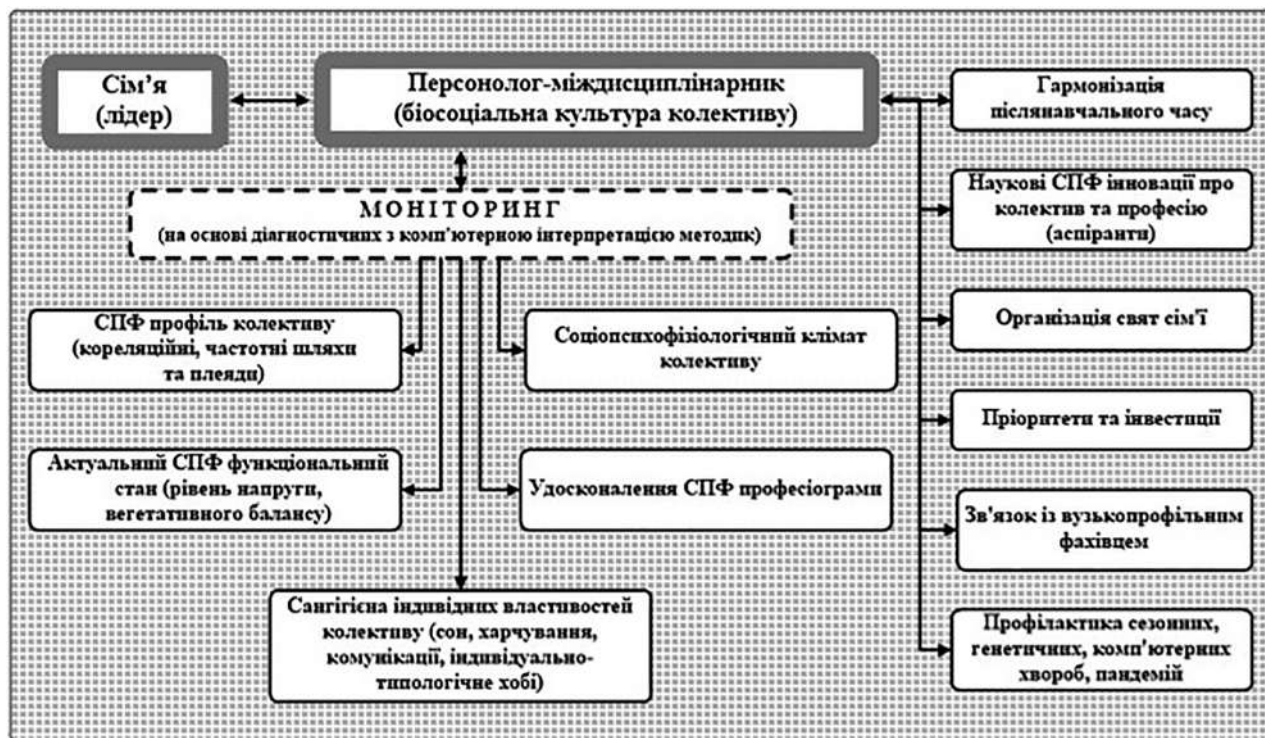


Рис. 9 Функціональні зв'язки між компонентами сімейного колективу

- **Особливо цінним у методології проблеми «Сімейна медицина» є «спільне ухвалення рішення» - процес, у якому бере участь пацієнт, вирішуючи з лікарем всі спірні питання щодо власного здоров'я.**

- **Індивідуально-типологічний підхід** до тандему пацієнт-лікар дуже важливий у плані поліпшення симбіотичних відносин, набуває ще більшого значення, ніж партнерство. Це вже взаєморозуміння пацієнта та лікаря, вихід на якісно новий рівень відповідальності, нових знань клінічної практики у майбутньому. Основне завдання постає як адаптування цієї ідеї до розробленого раніше **СПФ модуля індивідуально-типологічного підходу** (а не середньостатистичного) (рис. 7, 8, 9).

- **Розглянуто комплекс методичних засобів** для отримання узагальнених СПФ характеристик портрету і сімейство кривих по ММРІ, розподіл типів особистості (холерик, флегматик, меланхолік, сангвінік, центр); організація комп'ютерного накопичення інформації про життєзабезпечення сім'ї, спостереження за медичними протоколами здоров'я.

- **Розглянуто план електронної бібліотеки** з людинознавства і людиноведення (СПФ), що складається з інформаційно-довідкової та аналітичної систем. **Перша система** накопичує СПФ інформацію (спрощену) про сім'ю, **друга** – для наукової переробки спеціалістами про СПФ механізми індивідуальної адаптації сім'ї, оптимізацію життєзабезпечення, вікової норми, профілактики факторів ризику, довголіття, гено-фенотипових співвідношень показників.

- **Розроблено схему комп'ютерної структури СПФ обслуговування сім'ї на дому.** Розглянуті основні блоки спектру медико-санітарної допомоги сім'ї, вибір невеликих команд, медичний радник, телеконсультації (online).

- **Оновлено і суттєво доповнено структуру СПФ протоколу здоров'я із інформаційно-ємними показниками організму:** структурно-функціональна організація головного мозку, особливості обміну речовин, рівень інтоксикації організ-

му, спрямованість дезадаптації, бактеріо-, метеочутливість, СПФ норми показників різних вікових груп.

- **Оформлено блок адаптивного моніторингу властивостей і станів колективу на базі експрес-діагностичних методик** (індивідуальні і групові СПФ портрети, психометрія, активометрія, соціометрія та інше).

- Розроблено комплекс цілісних цифрових інноваційних СПФ показників (максимальний кореляційний шлях, частотні плеяди, періодограми здоров'я, професіограма; в тому числі показники системної реакції судомної активності, що відбиває дисгармонію саморегуляції організму).

- **Коротко розглядається величезний пласт знань** (з домінуванням власне авторських) про структуру особистості в сучасній персонології, функціональний стан організму, біосоціальну культуру, принципи питання системного підходу в людиноведенні (Казначєєв, Дільман, Богомолець, Амосов, Анохін, Глушков, Бехтерева, Собчик та інші), спрямованих на пошуки СПФ механізмів управління цілісністю здоров'я.

- **Особливо ж важлива управлінська компетентність, СПФ знання механізмів взаємодії та взаємосодії, біоритмологічного статусу колективу; біонічного, біокибернетичного відродження, подальших досягнень знань у людиноведенні для СПФ оптимізації життєзабезпечення світу.**

- **Універсальний СПФ модуль (технологія)** тепер вже зі здоров'язберегаючою підтримкою «Сімейна медицина на дому» дуже доречний під час кластеризації як форми реалізації інноваційних наукових розроблень НАН України, як очевидного резерву економічного та культурного потенціалу не тільки України, а всієї світової спільноти.

7. Загальні висновки

Грамотне, коректне управління людиною, а тепер вже й усім світом, починається зі знання механізмів її адаптації до різних екзо- та ендогенних факторів: мікро- (бактерії та віруси) та макро- (асоціальна та антисоціальна поведінка людини). Особли-

во важлива управлінська компетентність, знання СПФ механізмів людської взаємодії та взаємодопомоги.

В даний час дозрів фундамент для цілісного, системного міждисциплінарного розуміння людини, особливо професіонала і колективу в цілому на підставі індивідуально-типологічного (а не середньостатистичного) підходу з такими високоранговими якостями як емпатія (почуття співпереживання), мистецтво компромісу і взаємодії, стратегічне та тактичне мислення, раціональність, швидкодія, творчий, реалізаційний і контролюючий потенціал. Духовність і інтелігентність, постійна вдосконаленість інтелекту і віддача його результатів соціуму розглядаються як вектори успішних гуманістичних і практичних реалізацій.

Вирішення необхідної суперактуальної проблеми всього світу – СПФ (соціопсихологічна) людина, вимагає суттєвої інформаційно ємної та дуже своєчасної комп'ютерної підтримки.

Сьогодні наука в усьому світі логічно підійшла до етапу міждисциплінарного характеру знань про людську індивідуальність та її компоненти (динамічна структура особистості, її індивідуальний адаптаційний інформаційний простір - ІАП), системотворча синхронізація в живих системах (нелінійність), головний мозок - основна управляюча ланка організму, системоутворююче ядро цілісності людини, провідний лейтмотив життєвої стійкості людини: її професійна компетенція - біосоціальна культура (СПФ підтримка).

Педагогіка як наука про виховання, освіту та практичну реалізацію професії сьогодні повинна розглядатися в *послідовній єдності*, ґрунтуючись на численних знаннях, у тому числі про людину та людство.

Як резюме науково-освітня сфера має бути престижною, а здобуття нових знань – ключовим фактором для функціонування держави та її стратегічної безпеки. Національна ідея кожної держави має бути престижною і базуватися на уміннях, здібностях і талантах людини як справжній потужний резерв її благополуччя.

Водночас, як і раніше, важливі основні концептуальні положення системно-

го погляду В.І. Вернадського на зміст і суть академічної науки в незалежній державі в інформаційному періоді (як і в індустріальному) - як головного наукового центру, що поєднує державні дослідні установи з широким спектром технічних, природничих та соціо-гуманітарних наук.

Для більшої загальної впорядкованості наукової інформації, для побудови всіляких інформаційних систем глобального чи локального значення представляє інтерес Карта науки [32], в якій виділяється вагома група схожих наук – кібернетика, системологія, інформатика, економіка, математика та звісно ж біоніка, еволюція якої вже давно розкрила свої СПФ таємниці (їх треба лише побачити).

Комп'ютерна техніка перенапружена, біотехнології чекають на технологічні алгоритми для реалізації важливих інноваційних сервісно ефективних засобів та способів виробництва з високим кількісним і якісним потенціалом.

Сказане відкриває шлях для вдосконалення структур систем, оптимізації їхнього функціонування, тиражування типових проектних рішень, реалізації банків наукових даних, їхнього широкого практичного використання.

Розглянута тріада систематизованих та уніфікованих СПФ знань (Універсальна технологія СПФ модуля, «Сімейна медицина на дому», Комп'ютерна система діагностики та управління СПФ функціями колективу) повинна реалізуватися в майбутньому як принципово нова, багатогранна з оптимізацією і синхронізацією процесів колективу, з моделюванням, обробкою складних подій, прогнозуванням, візуалізацією, максимальним наближенням до автоматизації управління, що може стати найкращим сучасним засобом підвищення економічного потенціалу країни та оптимізації її біосоціальної культури.

У ще більшому практичному аспекті нова медична СПФ технологія формулює принципово нову матрицю здоров'я для вдосконалення тандему «Професійна компетентність – біосоціальна культура спеціаліста», підсилення знань лікарів, особливо сімейних; коректне керування станом па-

цієнта з урахуванням знань про квантову V-образну організацію інформаційного потоку, системної організації функцій, знань про їхню ритмо- та типологічну архітектуру; при цьому реалізується автоматична СПФ діагностика колективів (особливо експрес-реалізація, психоклімат, напруга, соціальність та інше); розширення можливостей дружніх інтерфейсів в аспекті СПФ онлайн-програм, профілактика сезонних захворювань (особливо при пандеміях), інтелектуалізація сервісно-функціональних можливостей комп'ютерів, участь у вдосконаленні цифрових технологій в медицині, що розростається. Технологічно-програмне забезпечення для майбутньої комп'ютерної системи діагностики та управління властивостями та станами людини докладно описана авторами [33,34,35].

Література

- Новіков Б.В. Творчість як спосіб здійснення гуманізму. Монографія. К.: Видавець ПА-РАПАН. 2005. 332 с.
- Букалов А.В. Интегральная соционика, типы коллективов, государств, этносоционика //Соционика, ментология и психология личности. Киев. 1998. №5. С. 13-17.
- Каганець І. Українська модель менеджмента //Персонал. 1999. С. 35-47.
- Кузьміна К.І. Дослідження функціональної організації головного мозку кролика під час переходу від неспання до наркотичного стану. Автореф. дис. канд. біол. наук. Кишинів. 1982. 16 с.
- Кузьміна К.І. Психофізіологічні механізми індивідуальної адаптації організму при дії різних екзогенних факторів. Дис. докт. біол. наук. Київ. 1995. 356 с.
- Удовин С.Л., Удовина О.Л. «Глобализация». Семиотические подходы. Образовательная библиотека. Рефл. - Бук. Ваклер. 2002. 461 с.
- Бондарев А.Г., Мажолис Е.Г. Гомеостатические взаимодействия человека, общества, природы и мышления //Гомеостатика живых, технических, социальных и экономических систем /Горский Ю.И., Астафьев В.И., Казначеев Л.П. Новосибирск. Наука. Сибирское отделение. 1990. 350 с.
- Дизайн. Иллюстрированный словарь-справочник. Г.Б. Минервин, В.Г. Шемко, А.В. Ефимов и др. Общая редакция Г.Б. Минервина и В.Г. Шемко. М.: Архитектура. 2004. 228 с.
- Кузьміна К. І., Сьомик Т. М. Здоров'язбеігаючі технології та комп'ютерні системи в сучасному людинознавстві для раціональної організації життєвого циклу людини //Проблеми програмування. 2017. № 3. С. 172-193. Режим доступу: http://nbuv.gov.ua/UJRN/Progr_2017_3_13.
- Кузьміна К.І., Сьомик Т.М., Погоріла І.О., Козак Н.С. Розробка теоретичних засад для сучасної комп'ютеризованої організації трудового колективу //Наукове оточення сучасної людини. Книга 4. Частина 1, розділ 1: серія монографій. Одеса: Купрієнко С.В. 2021. 17-55 с. ISBN 978-617-7880-15-7. <https://www.sworld.com.ua/simpua17sual17-1.pdf>.
- Копець Л.В. Психологія особистості. Навч. посіб. для студентів вищ. навч. закл. 2-е вид. К.: Вид. дім «Києво-могилянська академія», 2008. 458 с.
- Хьелл Л., Зиглер Д.. Теории личности. 3-е изд. СПб: Питер. 2013. 607 с. (Серия «Мастера психологии»).
- Теплов Б.М. Новые данные по изучению свойств нервной системы человека. Типологические особенности высшей нервной деятельности. М.: Медицина. 1963. Т. 3. С. 34-46.
- Небылицин В.Д. Психофизиологические исследования индивидуальных различий. М.: Наука. 1976. 336 с.
- Суворова В.В. Психофизиология стресса. М.: Педагогика. 1975. 208 с.
- Аболин Л.М. Психологические механизмы эмоциональной устойчивости человека. Казань.: Изд. Казанского университета. 1987. 262 с.
- Eysenck H.S. The biological basis of personality. Illinois. Springfield, 1967. 202p.
- Собчик Л.М. Психология индивидуальности. Теория и практика психодиагностики. СПб.: Речь. 2008. 624 с.
- Леонгард К. Акцентуированные личности. Киев: Вища школа. 1981. 270 с.

20. Ливанов М.М. Пространственная организация процессов головного мозга. М.: Наука. 1972. 260 с.
21. Коган А.Б. О принципах нейрональной организации рабочих механизмов управления функциональной системой // Принципы системной организации функций. М.: Наука. 1973. С. 125-130.
22. Сухомлинов А.И. Цифровые технологии и трансформация промышленных предприятий // Научное окружение современного человека. Книга 4. Часть 1. // Серия монографий. Одесса.: Куприенко С.В. 2021. С. 65-72.
23. Кузьміна К.І., Сьомик Т.М., Шереметова Л.М. Інноваційна педагогічна комп'ютерна технологія оптимізації шкільного колективу в аспекті знань про біосоціальну структуру особистості його членів // Проблеми програмування. 2019. № 4. С. 121-131.
24. Чадюк А.В., Федоренко Р.М., Курченко О.А. Розроблення та впровадження системи управління господарсько-майновим комплексом НАН України. // Проблеми програмування. 2023. № 2. С.24-39.
25. Путилов А.А. Системообразующая функция синхронизации в живой природе. Методологический очерк. Сибирское отделение. Новосибирск: Наука. 1987. 143 с.
26. Василевський Н.Н. Экологическая физиология мозга. М.: Медицина. 1979. 200 с.
27. Крыжановский Г.Н. Некоторые основные закономерности осуществления биологических процессов и их роль в патологии // Патфизиология. 1974. № 6. С. 3-15.
28. Сёмик Т.М., Кузьмина К.И. Динамика ритмической структуры векодвигательной реакции (ВДР) при изменении состояния человека-оператора от операционного покоя до стресса. Киев. 1983. Деп. в ВИНТИ 31.08.1983, № 6769-П-83. 44 С.
29. Симонов П.В. Мотивированный мозг. М.: Наука. 1987. 238 с.
30. Григорян Р.Д., Лябах Е.Г. Энергетическая теория адаптации и патофизиология системных болезней // Патфизиология. 2008. Т. 5, № 2. С. 87.
31. Фокин В.Ф. Энергетическая физиология мозга. Антидор. 2003. 288 с.
32. Леонов В.П. Карта науки // Наука і наукознавство. Свідоцтво про реєстрацію авторського права України № 21747 від 21.08.07 р. Офіційний бюлетень № 13, 2007.
33. Сёмик Т., Кузьмина К., Карпинка Е., Милютина Е. (2021). Учет социопсихофизиологического (СПФ) фактора – тренды развития и перспективы использования в ИТ, экономической и образовательной областях // Научный взгляд в будущее, 1(20-01), 19-34. <https://doi.org/10,30888/2415-7538.2021-20-01-007>.
34. Звіт з виконання наукової роботи «Розроблення теоретичних основ для сучасної комп'ютеризованої організації трудового колективу». Шифр теми № 3-9-17. 2017-2021 рр. ІПС НАН України. № держреєстрації РК- 0122U002281.
35. Гриценко В.І., Котова А.Б., Вовк М.І., Кіфоренко С.І., Белов В.М. Інформаційні технології в біології і медицині: Курс лекцій: Навчальний посібник. – Київ: Наук. думка, 2007.-25 арк.:ІЛ.

References

1. Novikov B.V. Creativity as a way of realizing humanism. Monograph. K.: Publisher PARAPAN. 2005. 332 p.
2. Bukalov A.V. Integral socionics, types of groups, states, ethnosocionics // Socionics, mentology and personality psychology. Kyiv. 1998. No. 5. pp. 13-17.
3. Kaganets I.. The Ukrainian model of management // Personnel. 1999. P. 35-47.
4. Kuzmina K.I. Study of the functional organization of the rabbit brain during the transition from wakefulness to the narcotic state. Autoref. thesis Ph.D. biological of science Chisinau 1982. 16 p.
5. Kuzmina K.I. Psychophysiological mechanisms of individual adaptation of the body under the influence of various exogenous factors. Diss. dr. biological of science Kyiv. 1995. 356 p.
6. Udovin S.L., Udovina O.L. «Globalization». Semiotic approaches. Educational library. Refl. - Beech. Wackler. 2002. 461 p.
7. Bondarev A.G., Mazholis E.G. Homeostatic interactions of man, society, nature and thinking // Homeostatics of living, technical, social and economic systems / Gorsky Yu.I., Astafiev V.I., Kaznacheev L.P. Novosibirsk The science. Siberian branch. 1990. 350 p.

8. Design. Illustrated dictionary-reference book. G.B. Minervin, V.G. Shemko, A.V. Efimov and others. General editing by G.B. Minervina and V.G. Shemko. M.: Architecture. 2004. 228 p.
9. Kuzmina K. I., Somik T. M. Health-preserving technologies and computer systems in modern humanities for the rational organization of the human life cycle // Programming problems. 2017. No. 3. P. 172-193. Access mode: http://nbuv.gov.ua/UJRN/Progr_2017_3_13.
10. Kuzmina K.I., Somik T.M., Pohorila I.O., Kozak N.S. Development of theoretical foundations for the modern computerized organization of the labor team //Scientific environment of modern man. Book 4. Part 1, Chapter 1: Monograph Series. Odesa: Kuprienko S.V. 2021. 17-55 p. ISBN 978-617-7880-15-7. <https://www.sworld.com.ua/simpua17suall17-1.pdf>.
11. Kopets L.V. Personality psychology. Education manual for university students education closing 2nd edition K.: Ed. house «Kyiv-Mohyla Academy», 2008. 458 p.
12. Kjell L., Ziegler D.. Theories of personality. 3rd ed. SPb: Peter. 2013. 607 p. (Series “Masters of Psychology”).
13. Teplov B.M. New data on the study of the properties of the human nervous system. Typological features of higher nervous activity. M.: Medicine. 1963. T. 3. pp. 34-46.
14. Nebylitsin V.D. Psychophysiological studies of individual differences. M.: Science. 1976. 336 p.
15. Suvorova V.V. Psychophysiology of stress. M.: Pedagogy. 1975. 208 p.
16. Abolin L.M. Psychological mechanisms of human emotional stability. Kazan: Publishing house. Kazan University. 1987. 262 p.
17. Eysenck H.S. The biological basis of personality. Illinois. Springfield, 1967. 202p.
18. Sobchik L.M. Psychology of individuality. Theory and practice of psychodiagnostics. SPb.: Speech. 2008. 624 p.
19. Leonhard K. Accentuated personalities. Kyiv: Vishcha school. 1981. 270 p.
20. Livanov M.M. Spatial organization of brain processes. M.: Science. 1972. 260 p.
21. Kogan A.B. On the principles of neuronal organization of working mechanisms for managing a functional system // Principles of systemic organization of functions. M.: Science. 1973. pp. 125-130.
22. Sukhomlinov A.I. Digital technologies and transformation of industrial enterprises // Scientific environment of modern man. Book 4. Part 1. // Series of monographs. Odessa: Kuprienko S.V. 2021. pp. 65-72.
23. Kuzmina K.I., Somik T.M., Sheremetova L.M. Innovative pedagogical computer technology for optimizing the school team in the aspect of knowledge about the biosocial structure of the personality of its members //Programming problems. 2019. No. 4. P. 121-131.
24. Chadyuk A.V., Fedorenko R.M., Kurchenko O.A. Development and implementation of the management system of the economic and property complex of the National Academy of Sciences of Ukraine. //Programming problems. 2023. No. 2. P.24-39.
25. Putilov A.A. System-forming function of synchronization in living nature. Methodological essay. Siberian branch. Novosibirsk: Science. 1987. 143 p.
26. Vasilevsky N.N. Ecological physiology of the brain. M.: Medicine. 1979. 200 p.
27. Kryzhanovsky G.N. Some basic patterns of biological processes and their role in pathology // Pathophysiology. 1974. No. 6. P. 3-15.
28. Somik T.M., Kuzmina K.I. Dynamics of the rhythmic structure of the eyelid motor response (VDR) when the state of the human operator changes from operating rest to stress. Kyiv. 1983. Dep. in VINITI 08/31/1983, No. 6769-P-83. 44 S.
29. Simonov P.V. Motivated brain. M.: Science. 1987. 238 p.
30. Grigoryan R.D., Lyabakh E.G. Energy theory of adaptation and pathophysiology of systemic diseases // Pathophysiology. 2008. T. 5, No. 2. P. 87.
31. Fokin V.F. Energy physiology of the brain. Antidor. 2003. 288 p.
32. Leonov V.P. Map of science //Science and science. Certificate of registration of copyright of Ukraine No. 21747 dated 08.21.07. Official bulletin No. 13, 2007.
33. Somik T., Kuzmina K., Karpinka E., Milyutina E. (2021). Taking into account the sociopsychophysiological (SPF) factor - development trends and prospects for use in IT,

- economic and educational fields // Scientific view of the future, 1(20-01), 19-34. <https://doi.org/10.30888/2415-7538.2021-20-01-007>.
34. Report on the implementation of the scientific work «Development of theoretical foundations for the modern computerized organization of the labor team.» Topic code number 3-9-17. 2017-2021 IPS of the National Academy of Sciences of Ukraine. Republic of Kazakhstan State Registration No. 0122U002281.
35. Gritsenko V.I., Kotova A.B., Vovk M.I., Kiforenko S.I., Belov V.M. Information technologies in biology and medicine: Course of lectures: Study guide. - Kyiv: Nauk. dumka, 2007.-25 sheets: IL.

Одержано: 10.05.2024

Внутрішня рецензія отримана: 21.05.2024

Зовнішня рецензія отримана: 26.05.2024

Про авторів:

¹*Кузьмина Клариса Іванівна,*
доктор біологічних наук,
провідний науковий співробітник.

Кількість наукових публікацій
в українських виданнях - понад 150.
Кількість наукових публікацій
в іноземних виданнях - понад 20.

¹*Сьомик Тетяна Михайлівна,*
кандидат біологічних наук,
старший науковий співробітник.
Кількість наукових публікацій
в українських виданнях - понад 120.
Кількість наукових публікацій
в іноземних виданнях - понад 20.

²*Андон Андрій Пилипович,*
Інженер програміст.

Місце роботи авторів:

¹Інститут програмних систем НАН України,
03187, м.Київ,
проспект Академіка Глушкова, 40.

²Інститут кібернетики НАН України,
03187, м.Київ,
проспект Академіка Глушкова, 40.

R.D. Grygoryan, A.G. Degoda, T.V. Lyudovyk, O.I. Yurchak

SIMULATING OF HUMAN PHYSIOLOGICAL SUPERSYSTEMS: INTEGRATIVE FUNCTION OF ORGANS SUPPORTING CELL LIFE

A quantitative model of fluids' dynamics (MFD) in the human body is created. Initially, MFD was realized as an autonomous C# software module (SM) functioning under given dynamic input characteristics. Later, SM was incorporated into our special software-modeling tool (SMT) capable of simulating the main modes of the human physiological super-system (PSS) providing cells' life. MFD describes mechanisms regulating long-term blood, lymph, total cells', and intercellular volumes. SMT simulates both intracellular and multicellular mechanisms providing cell energy balance despite casual dynamics of energy consumption rate. Multicellular mechanisms include complex systems controlling systemic and regional hemodynamics, interaction of the liver with the pancreas, blood filtration in kidneys, bladder function, and liquid expirations in lungs and skin in the background of a dynamic external environment. The latter is a gas atmosphere with altering pressure, illumination, temperature, humidity, and wind speed. Models have been tested using algorithms that design scenarios, including simulation of either short-time or long-time (hours or days) observations. Input data include different combinations of internal and external parameters including osmotic, and oncotic pressures. Output data include the main parameters characterizing organs and life support systems. Both student-medics and physiologists interested in providing theoretical research can be users of SM.

Keywords: physical health, body fluids, physiological control, quantitative model, simulator

Р.Д. Григорян, А.Г. Дегода, Т.В. Людовик, О.І. Юрчак

СИМУЛЯТОР ФІЗІОЛОГІЧНИХ НАДСИСТЕМ ЛЮДИНИ: ІНТЕГРАТИВНА ФУНКЦІЯ ОРГАНІВ ЖИТТЄЗАБЕЗПЕЧЕННЯ КЛІТИНИ

Створено кількісну модель динаміки рідин (PPP) в організмі людини. Спочатку MFD був реалізований як автономний програмний модуль C# (SM), що функціонує при заданих динамічних вхідних характеристиках. Пізніше SM було включено в наш спеціальний інструмент програмного моделювання (SMT), здатний моделювати основні режими фізіологічної суперсистеми людини (PSS), що забезпечує життєдіяльність клітин. MFD описує механізми довготривалої регуляції крові, лімфи та загального внутрішньоклітинного та міжклітинного об'ємів. SMT моделює як внутрішньоклітинні, так і багатоклітинні механізми, що забезпечують енергетичний баланс клітини, незважаючи на випадкову динаміку швидкості споживання енергії. Багатоклітинні механізми включають складні системи контролю системної та регіональної гемодинаміки, взаємодії печінки з підшлунковою залозою, фільтрації крові в нирках, функції сечового міхура, виділення рідини в легенях і шкірі на тлі динамічного зовнішнього середовища. Останнє являє собою газову атмосферу зі змінним тиском, освітленістю, температурою, вологістю та швидкістю вітру. Моделі були перевірені з використанням алгоритмів, які розробляють сценарії, включаючи симуляцію короткочасних або тривалих (години або дні) спостережень. Вхідні дані включають різні комбінації внутрішніх і зовнішніх параметрів, включаючи осмотичний і онкотичний тиск. Вихідні дані містять основні параметри, що характеризують органи та системи життєзабезпечення. Користувачами SM можуть бути як студенти-медики, так і фізіологи, зацікавлені у проведенні теоретичних досліджень.

Ключові слова: фізичне здоров'я, рідини організму, фізіологічний контроль, кількісна модель, тренажер

Introduction

Physiologists and physicians use empirical technologies to measure a limited number of life parameters that can quantitatively characterize the level of human physical health (HPH). Certainly, these parameters were statistically verified for populations of healthy and sick people. However, physicians meet a problem: the parametric landscape of HPH displays essential and often very complex alterations in time [1-3]. The origin of the observed instability of life parameters is still unknown. Moreover, neither physiologists nor medical scientists have ideas for conceptual overcoming the problem. In our opinion, the concept of physiological supersystems (PSS) [4-6], properly realized as a specialized computer simulator, could essentially assist in finding a way out of this methodological dead-end. The simulator must be based on special quantitative models of organs and anatomical-physiological systems that, according to PSS, were saved in vertebrates due to the essential role of these organs in evolution. Basic mathematical models and software components have been already developed [7-10].

The article proposes our vision of the final model, which integrates these components. Additionally, the article presents several test simulations demonstrating the potential of our software for physiologists and physicians.

Mathematical model of PSS

According to the concept of PSS, optimal cell metabolism is a fundamental condition for the cellular long life and functionality in unstable external/internal environments. Interactions between certain biochemical factors (also known as adaptation factors [1,11,12]) of every specialized cell and its nucleus adapt the expression of genes responsible for the efficiency of many molecular events including the rate of energy (ATP molecules) synthesis. However, the efficiency of autonomous intracellular regulators essentially falls in parallel with an increasing imbalance between biosynthesis and molecular destruction. Therefore, special adaptation factors, produced in these problematic cells and penetrated lymph or intercellular space, finally modify the activity of

multiple multi-cellular mechanisms (MM). Some of MM enhances the basic intracellular regulators. In particular, a set of MM provides molecular rebuilding in the problematic cells with materials (oxygen, water, nutrients). Another group of MM purifies the cytoplasm in these cells. So, the entire organism is chronically under the influence of adaptation factors. This is the humoral mechanism capable of originating HPH's instability. As many chemicals influence target neurons of CNS, the latter also modifies the parametric landscape of HPH. It is the understanding that the landscape depends on the number of problematic cells.

Every cell is a dynamic object adaptively compensating for intracellular destructions. In this context, one of the roles of MM is providing intracellular adaptive re-buildings with adequate inflows of primary materials (nutrients, oxygen, and water). The second role of MM provided by other organs is in adequately removing metabolic wastes. To provide such a requirement, PSS must have a rather complex construction. It was shown [5,6] that the digestive system, lungs, cardiovascular system (CVS), liver, pancreas, kidneys, thermoregulatory system (TS), as well as special neural-hormonal mechanisms providing the fluids' homeostasis are components of the PSS to be modeled. The main structural-functional blocks of PSS and their interrelations are illustrated in Fig. 1.

According to Fig.1, body fluids are partially located in four main reservoirs: cellular, intercellular, cardiovascular, and lymphatic. The cardiovascular system (CVS) is the primary space for receiving the fluids absorbed in the gastrointestinal tract. A part of blood through capillaries penetrating the interstitial liquid space is the internal source for cells taking oxygen and nutrients. The blood, remaining in capillaries, flows into venous capillaries and veins finally filling the central vein which is the reservoir for filling the right heart. So, in statics, the blood volume that leaves CVS and lymphatic return to CVS are equal. However, multiple factors can violate this equality.

The interstitial reservoir plays a dual role. It is the second source of lymph and the space for the liquid exchanging with cells. The

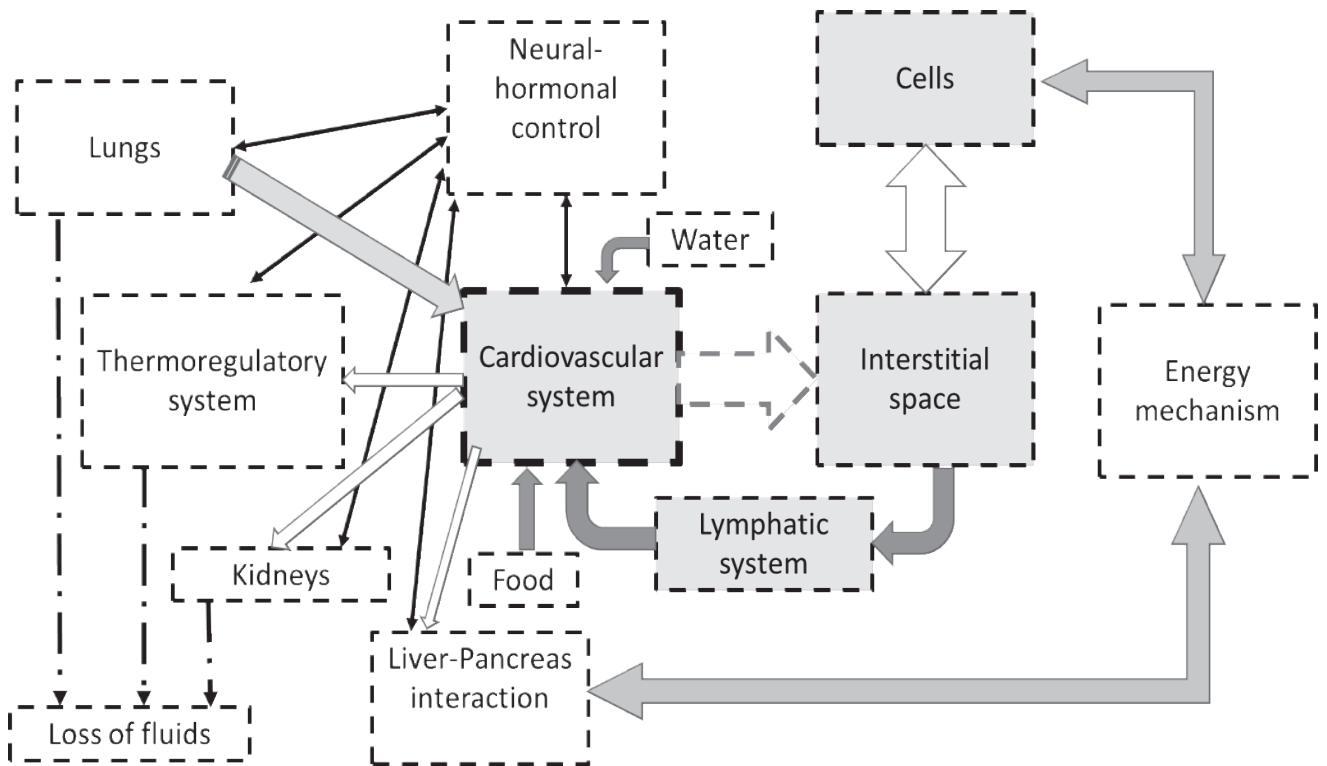


Fig. 1. Interaction of organs and anatomical-functional systems involved in PSS

cellular liquid contains both oxygen and nutrients. They are internal sources for the molecular synthesis of specific biological macromolecules. A part of them (molecules of ATP) provides all forms of intracellular biological work. Other macromolecules are ultra-structural units for constructing cellular organelles.

Thanks to liver-pancreas interaction, excess blood sugars can be transformed and accumulated into the liver as glycogen. Under low blood concentrations of glucose, this glycogen is back-transformed to glucose.

The dynamics of body fluids are associated with the activities of multiple specialized organs that are under the influences of nervous and hormonal regulators. Models of these mechanisms have been already published by us [7-10].

Fluids can leave the body in three main ways – blood filtration in the kidneys, skin sweating (this is under mechanisms controlling the thermoregulation), and lung expiration. Therefore, the total body fluids and their parts in the four relatively isolated internal spaces can display complex dynamics depending on multiple external and internal factors. Our models were properly advanced: the dynamics of internal fluids' redistributions are described.

Let us denote the body's total fluid volume for every time moment as $V_T(t)$. It can be presented as a sum of four components:

$$V_T(t) = V_{CVS}(t) + V_{VC}(t) + V_{In}(t) + V_{Ly}(t) \quad (1)$$

In (1), V_{CVS} is the total blood volume, V_{VC} is the total liquid into cells, V_{In} is the volume of the intercellular space, and V_{Ly} is the lymphatic volume.

$V_{CVS}(t)$ depends on the velocity of blood's inflows and outflows. Let us the water inflow is $q_{Imp}(t)$, $q_{LE}(t)$ is the outflow (expiration) in lungs, $q_{SE}(t)$ is the evaporation through the skin, $q_U(t)$ is the urine outflow, $q_{VC1}(t)$ and $q_{VCC}(t)$ are the flows of liquid into and out of a virtual cell, $q_{In}(t)$ is the inflow into the intercellular space, $q_{CF}(t)$ is the filtration in capillaries, and $q_{Ly}(t)$ is the lymph flow. Then, the differential equations (2)-(5) below describe relationships between volumes and liquid flows:

$$\frac{dV_{CVS}}{dt} = q_{Imp}(t) - q_{LE}(t) - q_{SE}(t) - q_U(t) - q_{CF}(t) - q_{In}(t) + q_{Ly}(t) \quad (2)$$

$$\frac{dV_{VIn}}{dt} = q_{Cf}(t) - q_{VCC}(t) - q_{Ly}(t) \quad (3)$$

$$\frac{dV_{VCC}}{dt} = q_{Cf}(t) - q_{Ly}(t) \quad (4)$$

$$\frac{dV_{VLy}}{dt} = q_{VCC}(t) - q_{Ly}(t) \quad (5)$$

The next system of equations (6)-(8) describes relationships between flows, pressures, and resistances in compartments:

$$q_{Cf}(t) = (P_C(t) - P_{In}(t)) / R_C(t) \quad (6)$$

$$q_{In}(t) = (P_{In}(t) - P_C(t)) / R_{InC}(t) \quad (7)$$

$$q_{Ly}(t) = (P_{Ly}(t) - P_{CV}(t)) / R_{Ly}(t) \quad (8)$$

In (6)-(8), $P_C(t)$ is capillaries' pressure, $R_C(t)$ is capillaries' resistance, $P_{In}(t)$ is interstitial liquid's pressure, $R_{InC}(t)$ is the resistance between interstitial and cell spaces, $P_{Ly}(t)$ is the lymphatic pressure, P_{CV} is the central vein pressure, and $R_{Ly}(t)$ is the resistance of lymphatic collector.

So, the equations' system (1)-(8) is incorporated into our algorithms developed for computer simulations of complex physiological mechanisms responsible for physical health. Calculations of pressures include the hydrostatic and osmotic (oncotic) components. Namely, two latter components play a key role in mechanisms that control the acid-basis and electrolytic homeostasis.

Brief information about research software

Simulation algorithms are similar to those published in [8,9]. Every simulation starts from the actualization of input data and simulation scenarios. These procedures are provided with assistant window forms. The simulation scenario covers both the simulation duration and combinations of dynamic tests. Our early software versions provided

the user with simulation graph results only after the entire scenario was processed. The calculation algorithm was modified to make simulations more effective without using a more empowered computer. To construct graphs, the modified algorithm creates and saves a series of data only for three neighbor time points (previous, current, next). This cardinally increased the real-time observation to be simulated. It can be both several seconds and several weeks. However, under long-lasting simulations, the user was not informed about the simulation dynamics which sometimes can be wrong. Thus the algorithm is modified to illustrate the calculation dynamics for a representative group of biometric characteristics that can inform the modeler whether the simulation goes in the planned course.

Main simulation results

The limited space of the article forced us to consider a limited number of test simulations. At the same time, we tried to illustrate both the simulation technology that may be interesting for programmers and several biological data that could attract physiologists and medics-researchers who are interested in widening the research methods.

This article does not provide the reader with simulation data reflecting the effects of concrete diseases. All simulations shown below have to demonstrate that our models and the software technology, despite being an interim product yet, are promising scientific tools. Perhaps, physiologists and medics-researchers as potential users of our software will know that after a proper modification, the proposed software can be used for mining additional information concerning functions of every organ involved in PSS.

Figures 2 and 3 illustrate special graph forms, providing tuning procedures for appropriate model components.

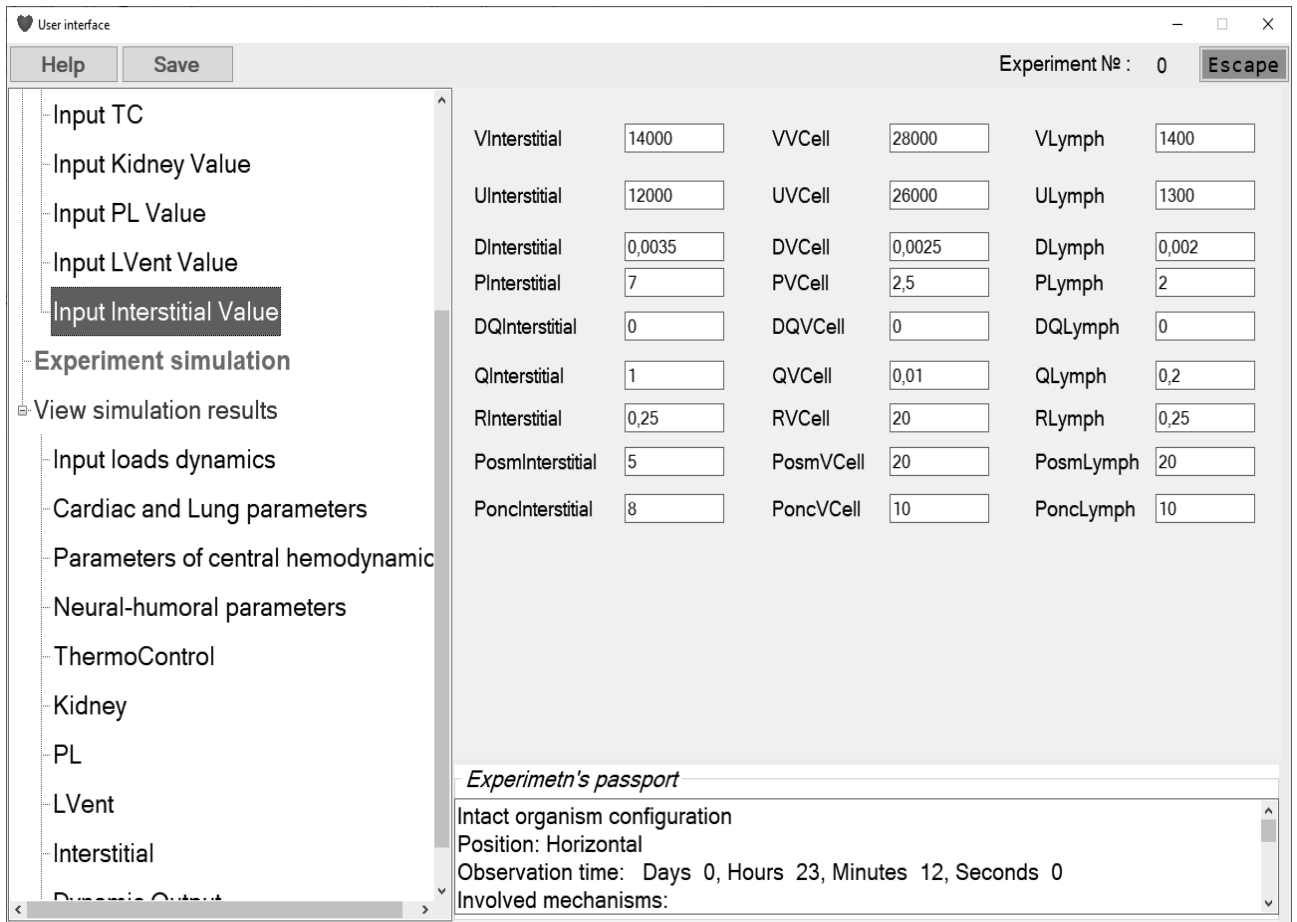


Fig. 2. A fragment of the user interface illustrating the window form for actualizing values of intercellular, cellular, and lymphatic compartments of the model before starting a simulation

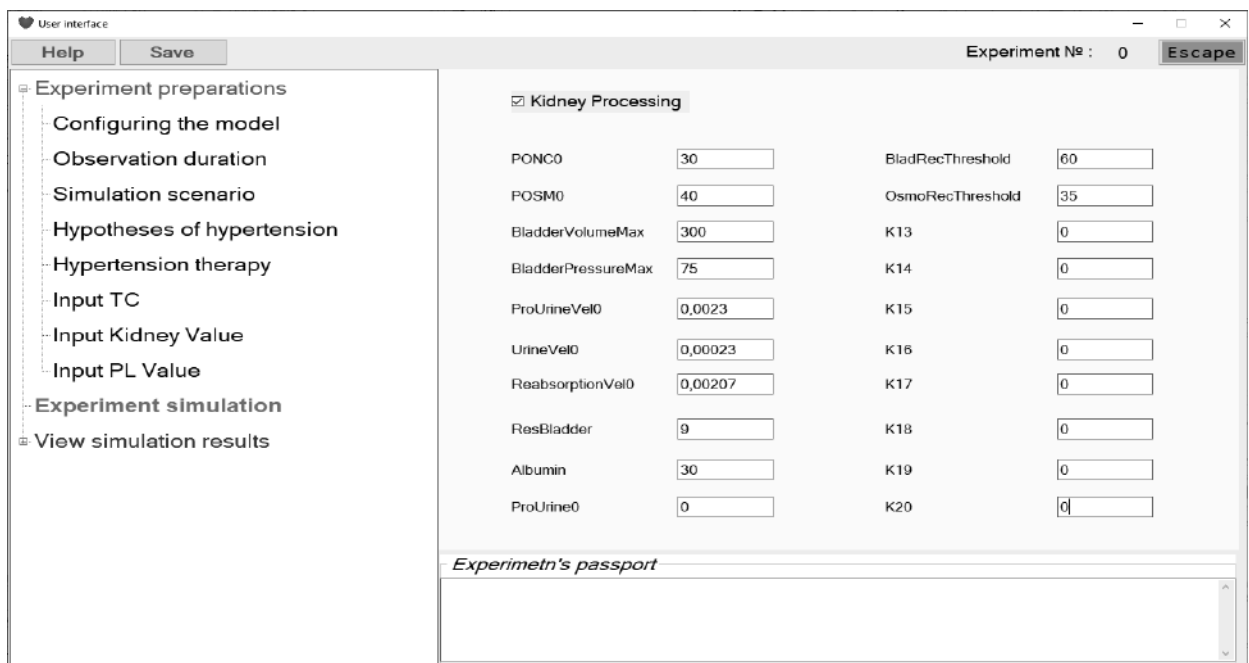


Fig. 3. A fragment of the user interface illustrating the window form for actualizing values of the kidney and bladder models before starting the simulation

Figures 4-12 illustrate several results of a computer simulation. The simulation scenario includes the following: 1) the human body position is horizontal; 2) neural and hormonal regulators controlling the function of CVS are switched on; 3) brain neurons reacting to alterations of liquor's osmotic pressure are switched off; 4) concentrations of atmosphere gases are in the norm; 5) within 24 hours period of observation the light intensity alters with the 12-hour day/night period starting from 6 hours of the morning; 6) air temperature, humidity, wind speed have the fixed dynamics; 7) serotonin and melatonin hormones display normal circadian dynamics.

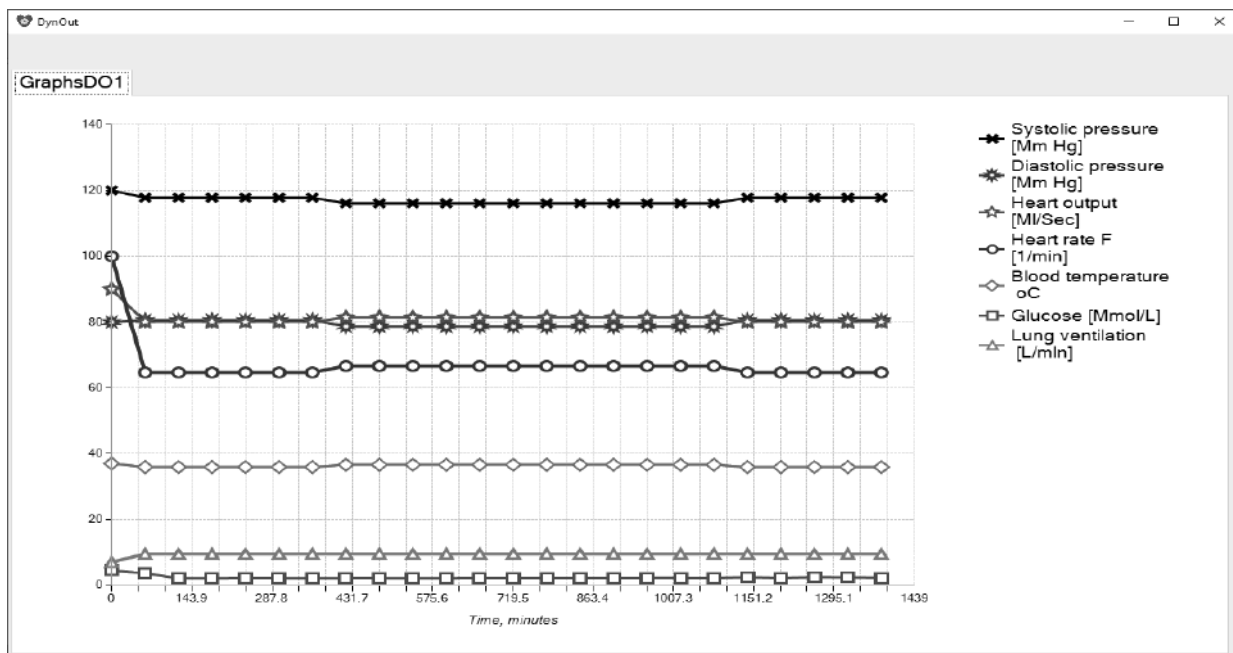


Fig. 4. Seven biological characteristics that can be specially displayed during calculations (the user arbitrarily sets intervals for every next time point)

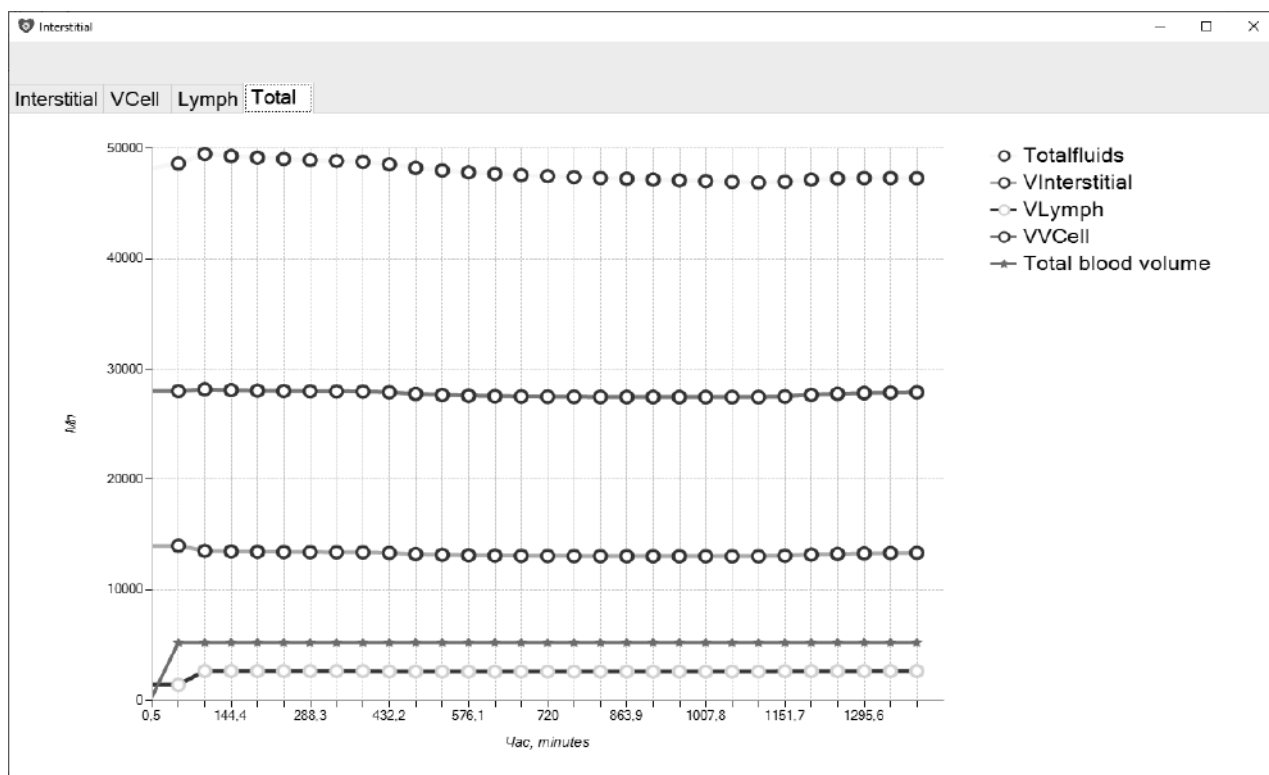


Fig. 5. The dynamics of body fluids (total and in four compartments). The simulation scenario provides a 24-hour experiment under natural dynamics of environmental temperature and lightness without activating the brain receptors of osmotic pressure

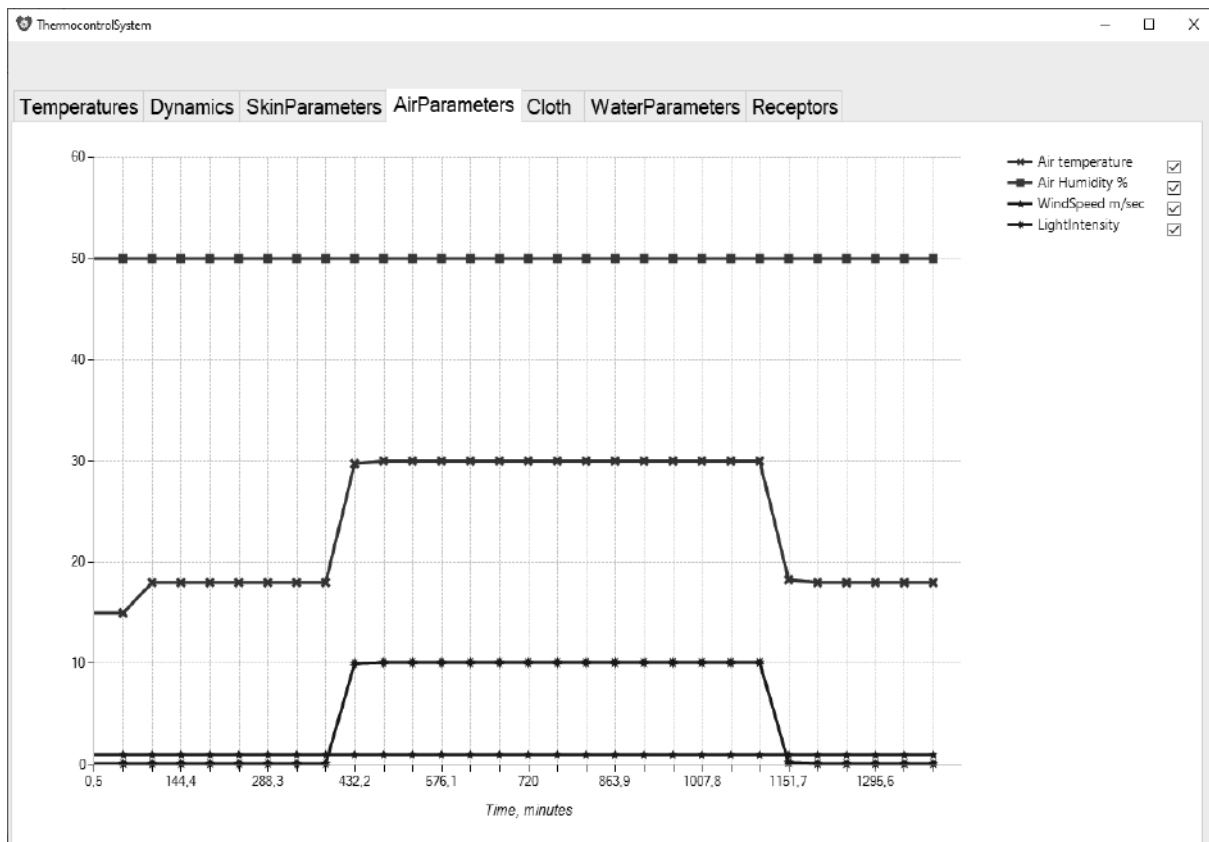


Fig. 6. Simulated dynamics of air parameters

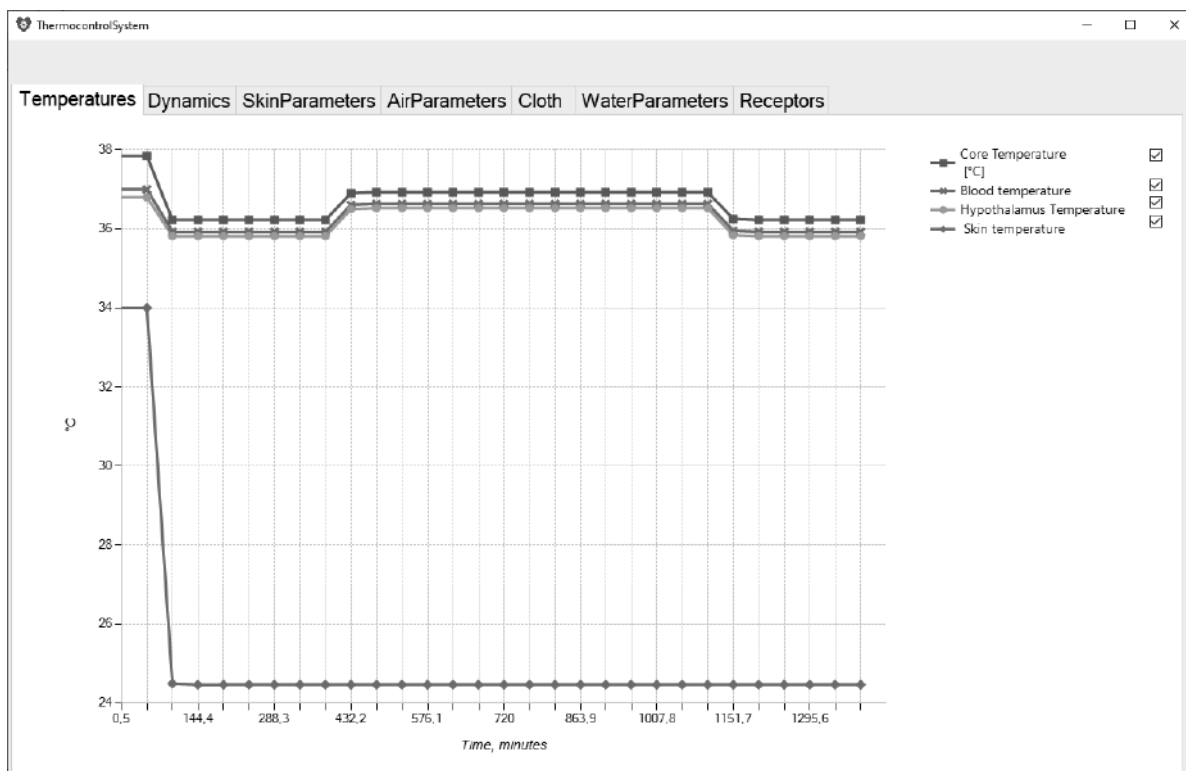


Fig. 7. Simulated dynamics of body temperatures according to the model of thermoregulation

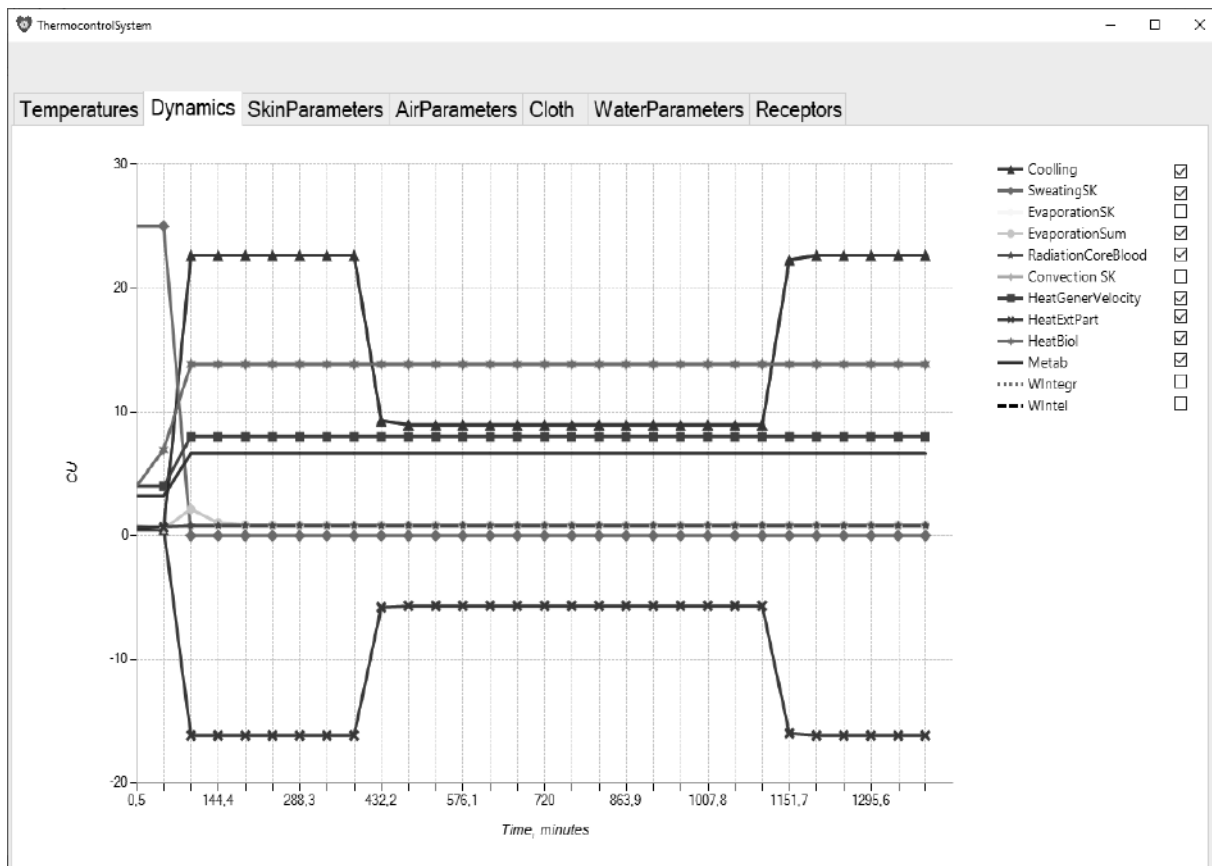


Fig. 8. Simulated dynamics of additional characteristics according to the model of thermoregulation

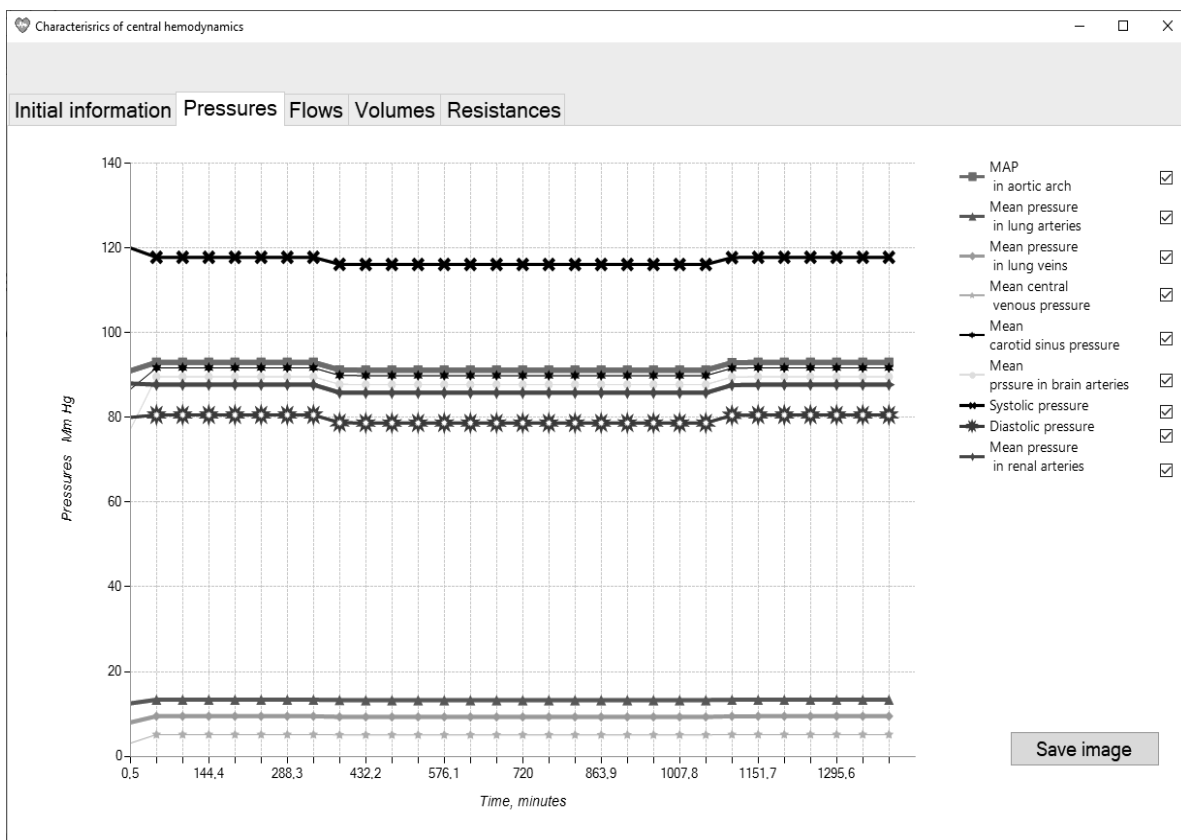


Fig. 9. Simulated dynamics of several cardiovascular characteristics

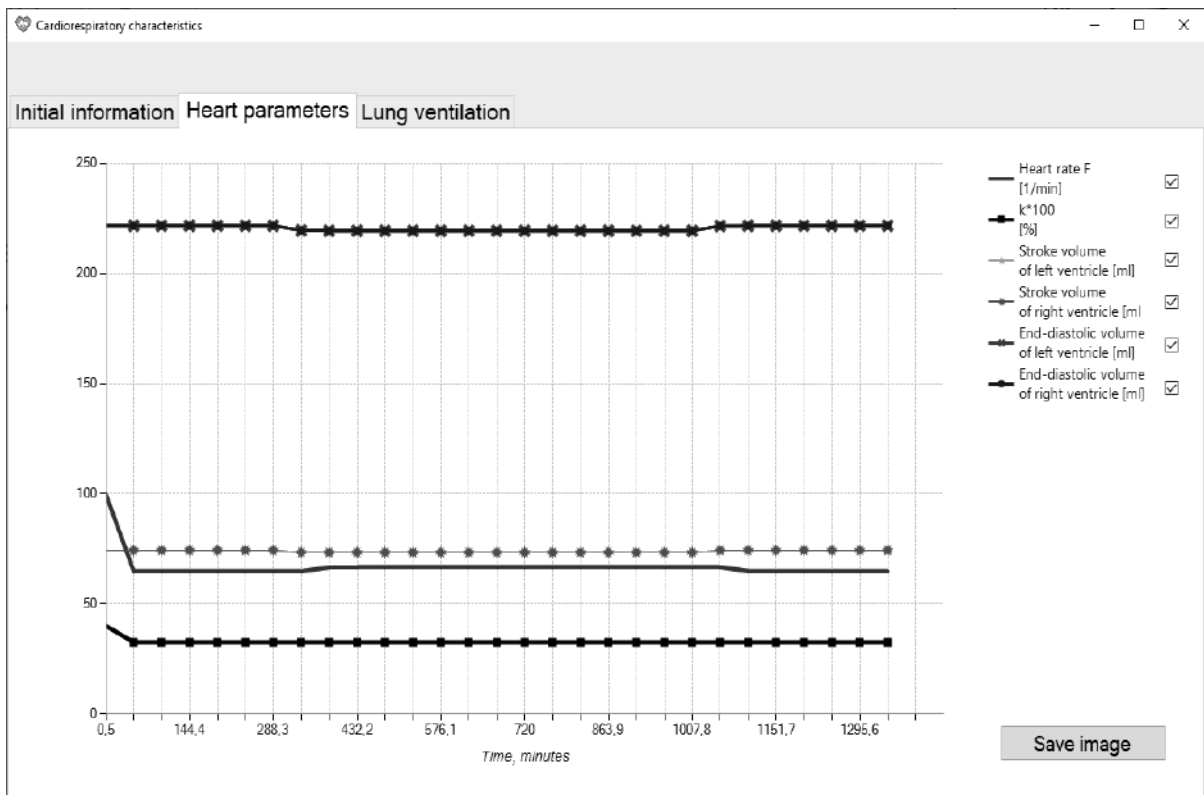


Fig. 10. Simulated dynamics of additional cardiovascular characteristics

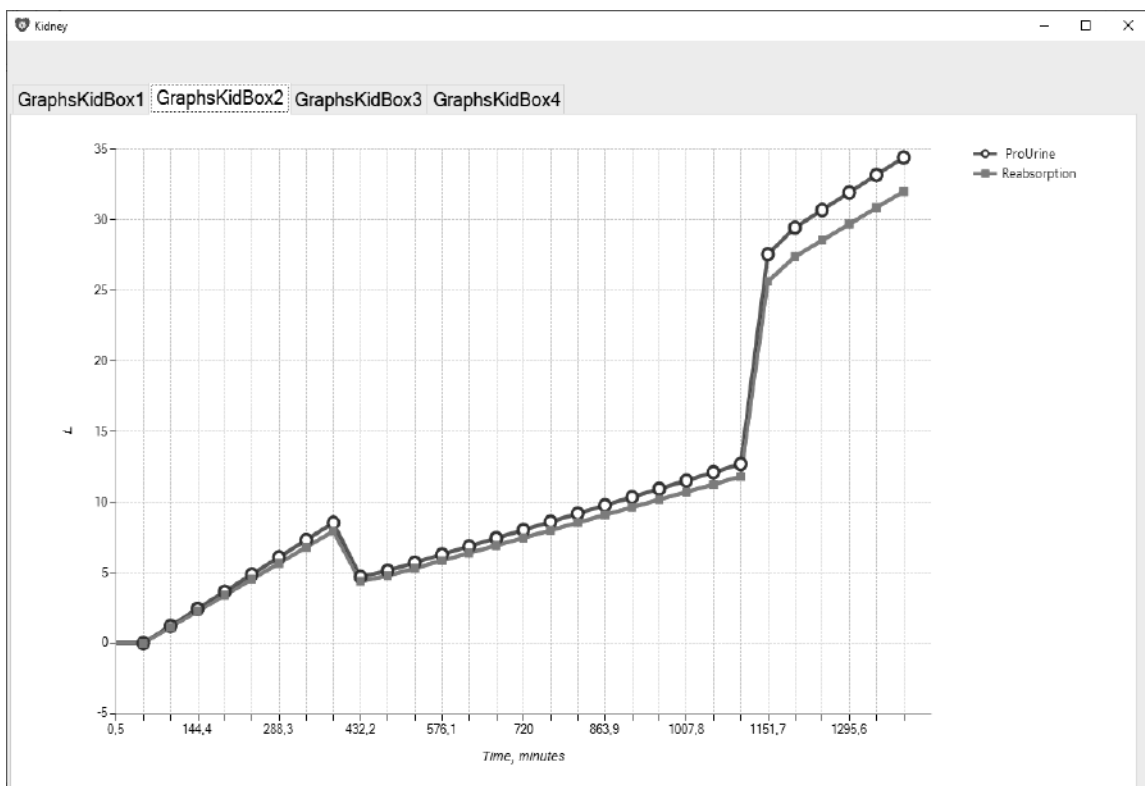


Fig. 11. Simulated dynamics of the kidney model demonstrating the difference between the production of the primary urine and its partial reabsorption

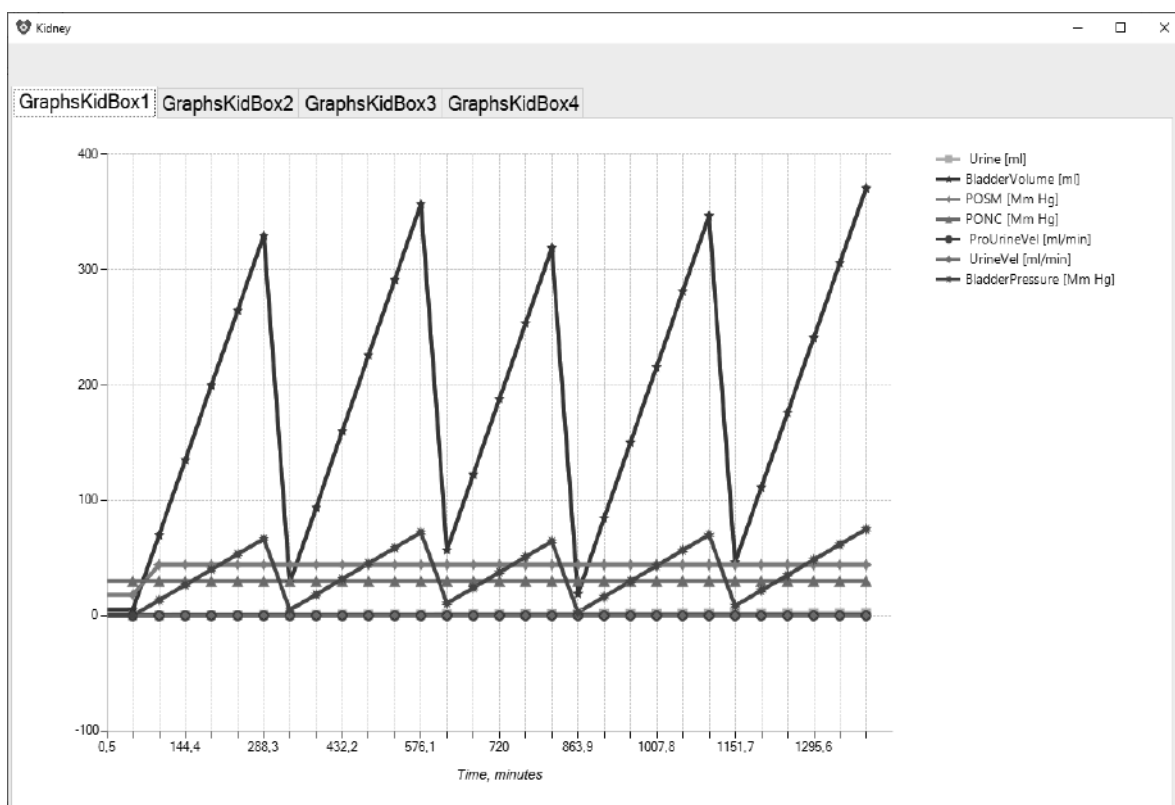


Fig. 12. Simulated dynamics of the kidney-bladder model demonstrating periodical fillings and emptying of the bladder per day (changes in hormonal activity taken into account)

Discussion

Our models are based on fundamental physiological knowledge including quantitative data like those presented in [2,3,11,12,16]. It was also critically analyzed the experience of colleagues [13-17]. Certainly, not every model constant is strongly verified. At the same time, such parameters were varied in some ranges to estimate their potential role in system-scale incorrectness. Results illustrated in Fig.4 – Fig.12 are only part of the information provided by our current software-modelling tool. It can simulate more thinkable scenarios and yield essentially more physiological output data. However, our current purpose was mainly to demonstrate the usability of the simulator. We plan to publish physiological aspects in biomedical journals. Before doing it, we aimed to thoroughly test every component model and the whole simulator using much more empirical data. Besides, the final user interface will be advanced to become maximally friendly for physiologists and physicians. We think that the fact that such a complex research tool has already been developed is an essential step forward.

Conclusion

For the first time, a special software-modeling tool (simulator) capable of essentially widening and deepening research opportunities of modern human physiologists and medics-researchers was developed. The main models, software units and the entire simulator had been mainly tested for the most well-known test scenarios. In addition to a big list of standard scenarios, the simulator provides the user with easy algorithms for constructing and simulating new scenarios. The simulator is the lonely research tool for obtaining novel data concerning the interaction of human organs that optimize cells' metabolism.

References

1. Grygoryan R.D. Extending the understanding of health mechanisms: inverse relationships between worsening of cells' metabolism and arterial pressure. *Znanstvena misel journal*, 2024, 89,35-43. Slovenia. DOI: 10.5281/zenodo.11049497.

2. Crystal G. J., Assaad S. I., Heerdt P.M. 24 - Cardiovascular Physiology: Integrative Function, Ed(s): Hugh C. Hemmings, Talmage D. Egan, Pharmacology and Physiology for Anesthesia (Second Edition), Elsevier, 2019, 473-519, ISBN 9780323481106, <https://doi.org/10.1016/B978-0-323-48110-6.00024-7>.
3. Eirin A, Lerman A, Lerman LO. Enhancing Mitochondrial Health to Treat Hypertension. *Curr Hypertens Rep.* 2018;20,10,89. doi: 10.1007/s11906-018-0889-4.
4. Grygoryan R.D., Sagach V.F. The concept of physiological super-systems: New stage of integrative physiology. *Int. J. Physiol. and Pathophysiology*, 2018: 9,2,169-180.
5. Grygoryan R.D. The optimal circulation: cells contribution to arterial pressure. N.Y.: Nova Science,2017: 287p. ISBN 978-1-53612-295-4.
6. Grygoryan R.D. The Optimal Coexistence of Cells: How Could Human Cells Create The Integrative Physiology. *J. of Human Physiol.* 2019,1 (01):8-28. DOI 10.30564/jhp.v1i1.1386.
7. Grygoryan R.D. Problem-oriented computer simulators for solving of theoretical and applied tasks of human physiology. *Problems of programming.* 2017, №3, 102-111.
8. Grygoryan R.D., Yurchak O.I., Degoda A.G., Lyudovyk T.V. Specialized software for simulating the multiple control and modulations of human hemodynamics. *Problems of programming.* 2021; 2: 42-53. DOI: <https://doi.org/10.15407/pp2021.02.042>.
9. Grygoryan R.D., Degoda A.G., Lyudovyk T.V., Yurchak O.I. Simulations of human hemodynamic responses to blood temperature and volume changes. *Problems of programming.* 2023; 1: 19-29. DOI: <https://doi.org/10.15407/pp2023.01.019>.
10. Grygoryan R.D. Modeling of mechanisms providing the overall control of human circulation. *Advances in Human Physiology Research*, 2022,4,5 – 21, <https://doi.org/10.30564/ahpr.v4i1.4763>.
11. Grygoryan R.D., Degoda A.G., Lyudovyk T.V., Yurchak O.I. Simulating of human physiological supersystems: interactions of cardiovascular, thermoregulatory and respiratory systems. *Problems of programming.* 2023, №3, P. 81-90. <http://doi.org/10.15407/pp2023.03.81>.
12. Shibuya M. Vascular endothelial growth factor and its receptor system: physiological functions in angiogenesis and pathological roles in various diseases, *The Journal of Biochemistry*, 2013,153, 1, 13 –19. <https://doi.org/10.1093/jb/mvs136>.
13. Camarda N, Travers R, Yang VK, London C, Jaffe IZ. VEGF receptor inhibitor-induced hypertension: emerging mechanisms and clinical implications. *Curr Oncol Rep.* 2022; 24:463–474. doi: 10.1007/s11912-022-01224-0.
14. Thomas S.R. Mathematical models for kidney function focusing on clinical interest. *Morphologie*, 2019, 103, Issue 343, 161-168. <https://doi.org/10.1016/j.morpho.2019.10.043>.
15. Sgouralis I, Layton AT. Mathematical modeling of renal hemodynamics in physiology and pathophysiology. *Math Biosci.*2015;264:8-20. doi:10.1016/j.mbs.2015.02.016.
16. Weinstein, A. M. Mathematical models of renal fluid and electrolyte transport: acknowledging our uncertainty. *American Journal of Physiology - Renal Physiology.* 2003, 284, 871–884.
17. Hardie D.G. Keeping the home fires burning†: AMP-activated protein kinase. *J. of The Royal Society Interface.* 2018,138. <https://doi.org/10.1098/rsif.2017.0774>.

Одержано: 27.09.2024

Внутрішня рецензія отримана: 01.10.2024

Зовнішня рецензія отримана: 15.10.2024

About authors:

Grygoryan Rafik,
Department chief, PhD, D-r in biology
<http://orcid.org/0000-0001-8762-733X>.

Degoda Anna,
Senior scientist, PhD.
<http://orcid.org/0000-0001-6364-5568>.

Lyudovyk Tetyana,
Senior scientist, PhD.
<https://orcid.org/0000-0003-0209-2001>.

Yurchak Oksana,
Leading software engineer.
<https://orcid.org/0000-0003-3941-1555>.

Place of work:

Institute of software systems
of Ukraine National Academy of Sciences
03187, Kyiv,
Acad. Glushkov avenue, 40,
Phone.: 066 683 0853.
E-mail: rgrygoryan@gmail.com,
anna@silverlinecrm.com,
tetyana.lyudovyk@gmail.com,
daravatan@gmail.com,

ВЛАСТИВОСТІ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПІД ЧАС УХВАЛЕННЯ РІШЕНЬ

У статті описується поняття функціональної стійкості інтелектуальних систем під час ухвалення рішень як одного з головних аспектів створення методів формалізації та моделювання знань і можливості застосування для підтримки ухвалення рішень в енергетиці. Сформульовані функції штучного інтелекту для створення інтелектуальних систем в енергетиці для підтримки ухвалення стратегічних рішень щодо розвитку енергетики з урахуванням вимог енергетичної безпеки. Використання принципів ситуаційного управління відповідає загальній схемі досліджень проблеми енергетичної безпеки та ухвалення стратегічних рішень, пов'язаних з оцінкою стану об'єктів енергетики та паливно-енергетичного комплексу загалом, а також із вибором основних напрямів їхнього подальшого функціонування та розвитку. У статті надано означення параметрів функціональної стійкості інтелектуальних систем під час ухвалення рішень та сформульовані основні характеристики надійності, стійкості та живучості інтелектуальних систем, представлено в графічному вигляді основні аспекти функціональної стійкості інтелектуальних систем у процесі ухвалення рішень, що відображають основні поняття ситуаційного керування, включно із ситуаційним аналізом і ситуаційним моделюванням з позиції дослідження проблеми енергетичної безпеки. Сформульоване поняття функціональної стійкості інтелектуальних систем дає можливість створювати алгоритми для подальшого використання під час розроблення програмного забезпечення для досліджень і підтримки ухвалення рішень в енергетиці.

Ключові слова: функціональна стійкість, надійність, живучість, відмово стійкість, штучний інтелект

G.V. Shuklin, O.V. Barabash, A.B. Grebennikov

FUNCTIONAL STABILITY OF INTELLIGENT SYSTEMS IN DECISION-MAKING

The article describes the concept of functional sustainability of intelligent systems in decision-making as one of the main aspects in the creation of methods for formalizing and modelling knowledge and the possibility of using it for decision support in the energy sector. The concept of functional stability was introduced for dynamic objects. However, for intelligent systems this concept is significantly different. This is due to the fact that the functioning of intelligent systems cannot be considered as the movement of an object. The functions of artificial intelligence for the creation of intelligent systems in the energy sector to support strategic decision-making on energy development, taking into account the requirements of energy security, are formulated. The use of the principles of situational management corresponds to the general scheme of research on the problem of energy security and strategic decision-making related to the assessment of the state of energy facilities and the fuel and energy complex as a whole, as well as the choice of the main directions of their further functioning and development. The article defines the parameters of functional sustainability of intelligent systems in decision-making and formulates the main characteristics of reliability, stability and survivability of intelligent systems, presents in graphical form the main aspects of functional sustainability of intelligent systems in decision-making, reflecting the basic concepts of situational management, including situational analysis and situational modelling from the perspective of studying the problem of energy security. The formulated concept of the functional sustainability of intelligent systems makes it possible to create algorithms for further use in the development of software for research and support of decision-making solutions in the energy sector.

Keywords: functional stability, reliability, survivability, fault tolerance, artificial intelligence

Вступ

Фундатором теорії функціональної стійкості є видатний Український вчений Машков О.А., який вперше ввів це поняття в теорії управління складними системами, навігації та управління авіаційно-косміч-

ними комплексами [1]. Функціональна стійкість об'єкта тісно пов'язана з такими властивостями як стійкість, надійність, живучість та відмовостійкість. Наразі отримали широкого застосування в управ-

лінні складними технічними системами інтелектуальні системи (ІС) підтримки ухвалення рішень (Intelligent decision support system). ІС - це системи підтримки ухвалення рішень, які широко використовують методи штучного інтелекту (ШІ). Штучний інтелект (ШІ) у технічних науках являє собою галузь комп'ютерних наук, яка створює програми і системи, здатні виконувати завдання, які зазвичай вимагають інтелектуальних здібностей людини. Ці завдання охоплюють сприйняття навколишнього середовища, навчання на основі досвіду, ухвалення рішень, розв'язання проблем, розуміння природної мови тощо. Сьогодні інтелектуальні системи ухвалення рішень отримують широке застосування в енергетиці.

В доповіді О. М. Суходоля [2] зазначається, що штучний інтелект в енергетиці застосовується для:

- **взаємодії з клієнтами** – чат-боти, які пропонують абоненту допомогу в отриманні наперед визначених стандартних відповідей або алгоритмів реагування;
- **управління мережею** – підвищення ефективності диспетчеризації та зменшення проблем з надійністю, що приводить до наявності необхідних експлуатаційних резервів. Крім того, за допомогою ШІ компанії спроможні оптимізувати роботу енергосистем за рахунок правильного розрахунку оптимального розподілу використання пропускної спроможності існуючих ліній передачі, а також визначення обладнання, яке може й надалі експлуатуватися або яке необхідно замінити;
- **інтелектуалізація мереж (Smart grids)** – оптимізація керування потоками енергії в системі між обладнанням споживачів (будівельні споруди), накопичувачами енергії (акумуляторами, батареями), відновлювальними джерелами енергії, мікромережею та центральною мережею;
- **мікромережа (Microgrid)** – невелика локальна енергетична мережа, яка працює незалежно від центральної мережі;
- **віртуальні електростанції (Virtual Power Plants)** – віртуальна енергосистема, яка незалежна від централізованої мережі і яка самостійно забезпечує балансуювання пропозиції та попиту між своїми учасниками;
- **створення нових бізнес-моделей роботи на ринку** – уникнення третьої сторони, а саме оператора системи передачі чи розподілу, у взаємовідносинах між виробником електроенергії та кінцевими споживачами;
- **торгівля електроенергією** – автоматичне створення угод про закупівлю електроенергії, що надає контрахтам більшої ефективності, скорочує час транзакцій та знижує вартість використання;
- **виявлення крадіжок електроенергії** – автоматичне виявлення аномалій у поведінці постачальників та споживачів із подальшим позначенням їх для детальної перевірки персоналом енергетичної компанії;
- **акумуляування енергії** – вирівнювання графіка навантаження системи і допомога в зменшенні потреби енергетичних компаній у будівництві нових електростанцій;
- **прогнозування режимів роботи інфраструктури та планування її розвитку** – спроможність енергетичної компанії передбачити час, коли енергоблок, трансформатор або окреме технічне обладнання мережі із високою ймовірністю вийдуть з ладу;
- **збільшення виробництва та будівництво** – на основі аналізу кліматичних та географічних умов допомога проєктувальникам вибрати кращі майданчики для будівництва вітрових, сонячних та термальних електростанцій;
- **підвищення рівня кібербезпеки енергосистеми** – виявлення та оперативне реагування на кібератаки;

- **системи управління виробничими процесами** – прогнозування попиту, оптимізація виробничих цілей, видача завдань для етапів технологічного процесу, виявлення подій, аномалій на виробничих лініях тощо.

Виходячи з вищевикладеного, виникає завдання визначення поняття функціональної стійкості інтелектуальних систем в ухваленні рішень з застосуванням ШІ як здатність складних технічних систем зберігати свої функції під час впливу деструктивних факторів.

Аналіз останніх досліджень. У роботі [3] здійснено аналіз сучасних стратегій розвитку існуючих інформаційно-енергетичних систем, а також огляд методів розробки та проблем теоретичних досліджень шляхів удосконалення інформаційно-енергетичних систем та мереж. Авторами введено поняття «розумної ефективності» і представлено топологію інформаційної мережі відповідно до концепції «Smart Grid» та топологію інтелектуальної енергосистеми, в якій не відображається система, яка б забезпечувала функціональну стійкість такої мережі. В роботі [4] здійснено аналіз технологічної концепції «Smart Grid» в електроенергетиці, представлено її структуру та функціональну схему електроенергетики на базі концепції «Smart Grid». Також відзначено основні параметри, які забезпечують надійність, як складової функціональної стійкості самої інтелектуальної системи. В роботі [5] визначено поняття кіберстійкості критичних інформаційних структур, що є однією із складових функціональної стійкості інтелектуальних систем. В роботі [6] розглядається стан та перспективи подальшого розвитку інтелектуального керування з представленням технологій та алгоритмом керування швидкісними динамічними процесами в розподіленому комп'ютерному середовищі та здійснено структурізацію моделей формування хаотичної поведінки інтелектуальної системи ухвалення рішень, а також створено методи покращення

якості обміну даними між компонентами мережевих систем. У роботі [7] запропоновано контекстну модель, яка відображає залежність між об'єктами, їх властивості та надає інформацію, необхідну для ухвалення рішень щодо подальших дій, однак ця модель не враховує наявності параметрів, які забезпечують функціональну стійкість самої ІС прийняття рішень. У роботі [8] для забезпечення функціональної стійкості інформаційних систем використовуються програмно-конфігуровані мережі, за допомогою яких відбувається відокремлення функцій передачі трафіку від функцій управління, що приводить до підвищення рівня безпеки функціонування інформаційних систем та дозволяє ввести в процес передачі пакетів поняття резерву часу. Однак в усіх дослідженнях або не розглядаються параметри, які забезпечують функціональну стійкість або розглядаються окремі компоненти функціональної стійкості інтелектуальних систем ухвалення рішень і відсутня загальна модель функціональної стійкості інтелектуальних систем, яка б інтегрувала в собі всі параметри її забезпечення.

Мета статті. Метою даної статті є визначення функціональної стійкості інтелектуальних систем у процесі ухвалення рішень енергетичного об'єкту.

Широке використання комп'ютерних систем підтримки ухвалення рішень сприймається як передача повноважень ухвалення рішень програмам. Це призвело до думки, що повноваження щодо ухвалення рішень можуть бути певною мірою делеговані особами, які ухвалюють рішення і надмірно централізовані, машинам, що активніше використовуються в циклі управління. Системи підтримки ухвалення рішень можуть складатися з таких компонентів: керування даними; керування моделлю; керування призначеним для користувача інтерфейсом і архітектура системи підтримки ухвалення рішень, як це схематично представлено на рисунку 1.

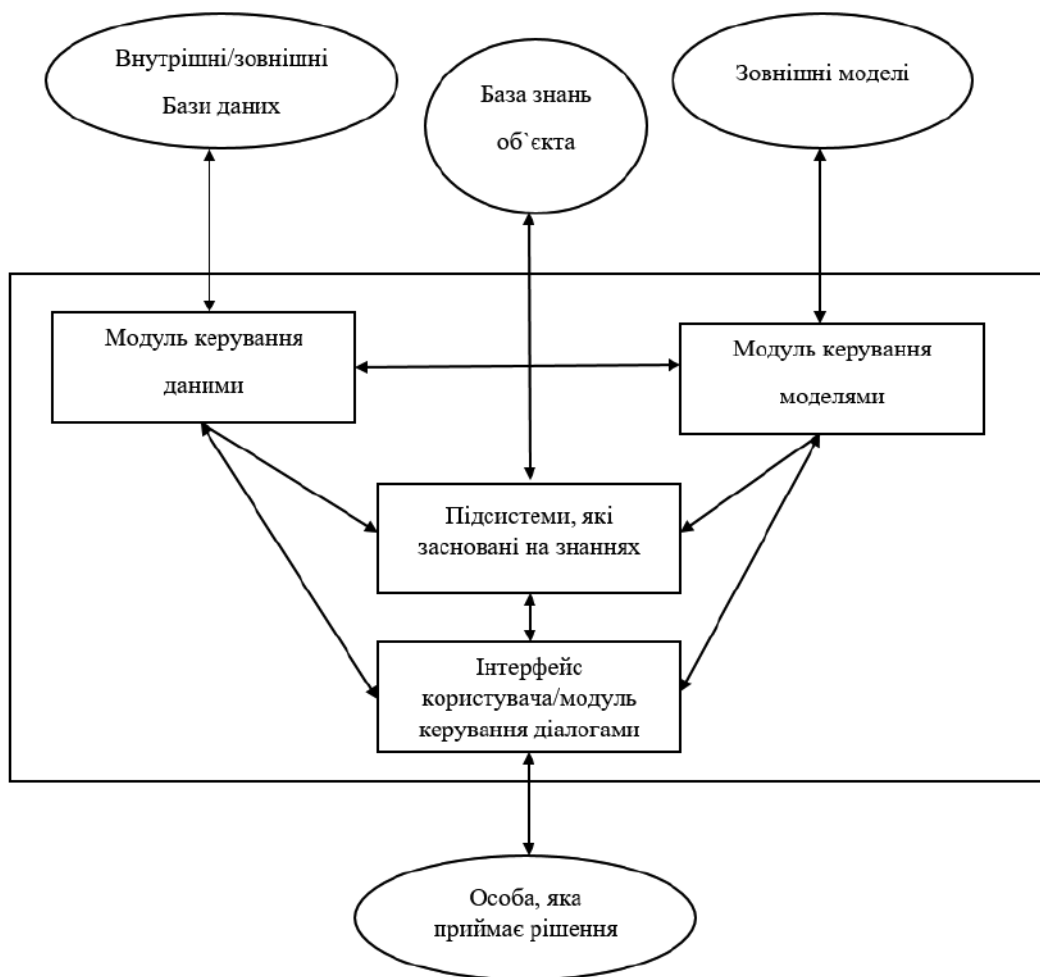


Рис. 1. Системи підтримки ухвалення рішень

Інтелектуальні здібності та поведінка, інтегровані з комп'ютерною системою, створюють інтелектуальну машину. Машина повинна виконувати функцію помічника в ухваленні рішень, пошуку інформації, управлінні складними об'єктами і, нарешті, розуміти значення слів. Щоб розробити інтелектуальну комп'ютерну систему, необхідно зібрати, організувати та використати людські експертні знання в деяких вузьких галузях; удосконалити обчислювальну міць мозку системи за допомогою складних алгоритмів, що використовують сенсорне опрацювання, моделювання світу, генерацію поведінки, оцінювання цінності та глобальну комунікацію; підрахувати кількість інформації та цінностей, які система зберігає у своїй пам'яті.

Крім того, інтелектуальна система має таку ознаку як здатність діяти відповід-

ним чином у невизначеному середовищі для збільшення ймовірності успіху, а успіх - це досягнення поведінкових підцілей, що підтримують кінцеву мету системи.

Для створення інтелектуальних систем в енергетиці можуть бути застосовані наступні функції ШІ:

функція прогнозування попиту на електроенергію – застосування алгоритмів машинного навчання для аналізу та прогнозування піків та попиту на електроенергію. Ця функція допомагає оптимізувати розподіл електроенергії та запобігати перевантаженням;

функція керування енергомережами – оптимізація роботи енергосистем та керування мережею, включаючи динамічне керування трансформаторами, маршрутизацію електроенергії та виявлення розладів;

функція оптимізації енергоспоживання – забезпечує адаптацію енергосистеми до зміну попиту на електроенергію, а також забезпечує регулювання попиту на електроенергію відповідно до параметрів, які забезпечують цю оптимізацію;

функція обслуговування та прогнозування відмов – застосування машинного навчання для аналізу даних від сенсорів та моніторингу стану обладнання з метою запобігання відмов та оптимізації планів технічного обслуговування;

функція енергетичного аналізу та оптимізація – застосування алгоритмів оптимізації та аналізу даних для пошуку оптимальних рішень в плані енергетичної ефективності, тобто оптимізація розподілу електроенергії в системі;

функція керування збереженням електроенергії – оптимізація процесів збереження електроенергії, включаючи прогнозування моментів часу найбільшого попиту та визначення оптимальних стратегій зарядження та розрядження;

функція моніторингу та запобігання аварій – виявлення аномалій, включаючи кібератаки, попередження про можливі аварії та надання рекомендацій щодо їх унеможливлення;

функція оптимізації енергетичних процесів у будівлях – керування системами опалення, вентиляції, кондиціонування повітря, освітлення та іншими системами енергоспоживання в будівлях з метою економії електроенергії.

Під час впровадження інтелектуальних систем ухвалення рішень на об'єктах енергетики важливо забезпечити стійкість функціонування самої інтелектуальної сис-

теми ухвалення рішень. Тобто необхідно визначити поняття функціональної стійкості інтелектуальної системи ухвалення рішень в енергетиці та її основні аспекти. В роботі [13] функціональна стійкість об'єкта визначається як його спроможність зберігати протягом заданого проміжку часу виконання своїх основних функцій в межах, встановлених нормативними вимогами, в умовах протистояння, а також впливів потоків відмов та збоїв.

Враховуючи те, що інтелектуальні системи являють собою особливий клас складних систем, то, не порушуючи загального поняття функціональної стійкості, для інтелектуальних систем під час ухвалення рішень, означення функціональної стійкості визначається наступним чином:

Означення 1. Функціональна стійкість інтелектуальних систем в ухваленні рішень – це їх здатність ефективно функціонувати та вирішувати завдання (забезпечувати безперебійне виконання функцій штучного інтелекту) в умовах різноманітних викликів, змін та непередбачуваних ситуацій.

Означення 2. Надійність інтелектуальних систем під час ухвалення рішень це їх здатність давати точні та надійні відповіді або рекомендації в різних ситуаціях. Це ключовий аспект, особливо в контексті використання штучного інтелекту в системах, де ухвалення правильних рішень є критичним.

У таблиці 1 представлено основні характеристики (параметри) надійності інтелектуальних систем під час ухвалення рішень.

Таблиця 1. Основні характеристики надійності інтелектуальних систем

Характеристика	Функція характеристики
Точність	Спроможність інтелектуальної системи надавати правильні відповіді або прогнози. Головною вимогою є високий ступінь точності в опрацюванні даних та алгоритмах ухвалення рішень.
Стабільність	Спроможність інтелектуальної системи проявляти стабільну поведінку в різних умовах і за різних типів вхідних даних і водночас відсутня висока чутливість до змін у даних або середовищі.

Узагальнення	Інтелектуальна система повинна бути здатна застосовувати отримані знання до нових, раніше не бачених ситуацій.
Прозорість і зрозумілість	Спроможність пояснити свої рішення: користувачі та фахівці повинні розуміти, чому інтелектуальна система ухвалила відповідне рішення. Це також допомагає у виявленні та усуненні можливих помилок.
Керування невизначеністю	Ефективне керування інтелектуальною системою невизначеністю, яка з'являється в даних, що обробляються, та в поточних умовах, включаючи спроможність обробляти неповні або нечіткі дані.
Безпека	Інтелектуальні системи мають бути захищені від втручання та атак, щоб запобігти спотворенню результатів або небажаним наслідкам.

Виходячи з характеристик, які представлено в таблиці 1, можна стверджувати, що надійність у контексті ШІ вимагає балансу між високою продуктивністю і точністю, а також умінням адаптуватися до різних сценаріїв використання. Відповідальне ставлення розробників включає ретельне тестування, забезпечення прозорості системи та безперервне оновлення і поліпшення моделей для підтримки їхньої надійності.

Здійснивши розкриття поняття надійності інтелектуальних систем під час ухвалення рішень, перейдемо до визначення поняття стійкості інтелектуальних систем в ухваленні рішень.

Означення 3. Стійкість інтелектуальних систем під час ухвалення рішень - це здатність зберігати надійність в умовах середовищ, що змінюються та в умовах середовищ, що атакують.

Стабільність є важливою властивістю інтелектуальних систем у процесі ухвалення рішень, оскільки системи можуть стикнутися з різними викликами під час експлуатації, включно з неочікуваними вхідними даними, атаками зловмисників, змінами в середовищі та іншими факторами.

В таблиці 2 представлено основні характеристики (параметри) стійкості інтелектуальних систем у процесі ухвалення рішень.

Таблиця 2. Основні характеристики стійкості інтелектуальних систем

Характеристика	Функція характеристики
Стійкість до вхідних даних	Спроможність ІС ефективно обробляти різноманітні, включаючи шкідливі дані, не допускаючи при цьому суттєвих збоїв або не коректних висновків.
Адаптивність	Здатність ІС адаптуватися до змін у зовнішніх умовах, включаючи до змін у даних, вимогах користувача або характеристиках середовища.
Стійкість до атак	Захищеність інтелектуальних систем від різних видів атак, таких як впровадження в систему, внесення спотворень у дані або впровадження шкідливого програмного забезпечення, включаючи заходи безпеки для запобігання і виявлення атак.
Спроможність відновлення	Спроможність ІС у разі збою або атаки бути здатною відновитися і продовжити свою роботу з мінімальними втратами.
Тестування та оцінка стійкості	Активне тестування ІС з боку розробників на стійкість та ефективне оцінювання її стійкості до різних сценаріїв.

Неперервне оновлення	Для підтримання стійкості, системи мають регулярно оновлюватися, включно з патчами безпеки і поліпшеннями в процесах ухвалення рішень.
-----------------------------	--

Стійкість інтелектуальних систем стає особливо важливою в контексті використання в енергетичному секторі, де навіть невеликий збій може призвести до серйозних наслідків.

Наступне поняття, пов'язане з функціональною стійкістю інтелектуальних систем під час ухвалення рішень є живучість.

Означення 4. Живучістю інтелектуальних систем під час ухвалення рішень - це здатність зберігати працездатність і ефе-

ктивність в умовах різноманітних збоїв, атак або інших несприятливих ситуацій.

Живучість охоплює не тільки здатність до відновлення після збоїв, а й активні заходи для запобігання, виявлення та мінімізації впливу негативних факторів на роботу системи.

В таблиці 3 представлено основні характеристики (параметри) живучості інтелектуальних систем під час ухвалення рішень.

Таблиця 3. Основні характеристики живучості інтелектуальних систем

Характеристика	Функція характеристики
Резервування та відмовостійкість	Спроможність ІС переключатись на альтернативні ресурси або методи у разі збоїв, включаючи в наявність резервних серверів, резервних алгоритмів або інших механізмів відмовостійкості.
Автоматичне відновлення	Здатність ІС самостійно відновлюватися після збоїв без необхідності втручання людини, включаючи автоматичне відновлення з резервних копій даних, перемикання на резервні канали зв'язку тощо.
Динамічне навчання та адаптація	Забезпечення ІС можливістю динамічного навчання на основі досвіду й адаптації до нових умов, що приводить до спроможності підлаштовуватися під обставини, що змінюються.
Моніторинг та виявлення аномалій	Здатність ІС активно здійснювати моніторинг свого стану та виявляти аномалії або нештатні ситуації для оперативного реагування на потенційні проблеми.
Гнучкість і маштабованість	Спроможність ІС бути гнучкими і здатними масштабуватися для ефективного керування мінливим обсягом даних або вимог.
Тестування сценаріїв збоїв	Спроможність ІС систематично тестувати на різні сценарії збоїв для переконання в їхній готовності до різних нештатних ситуацій.

Забезпечення живучості інтелектуальних систем є важливим аспектом у розробці та експлуатації, особливо в енергетиці, де надійність і доступність системи відіграють вирішальну роль.

Виходячи з наведених таблиць характеристик надійності, стійкості та живучості, можна визначити основні аспекти функціональної стійкості інтелектуальних систем під час ухвалення рішень, які наведено на рисунку 2.

**ОСНОВНІ АСПЕКТИ ФУНКЦІОНАЛЬНОЇ
СТІЙКОСТІ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ
ПРИ ПРИЙНЯТТІ РІШЕНЬ**

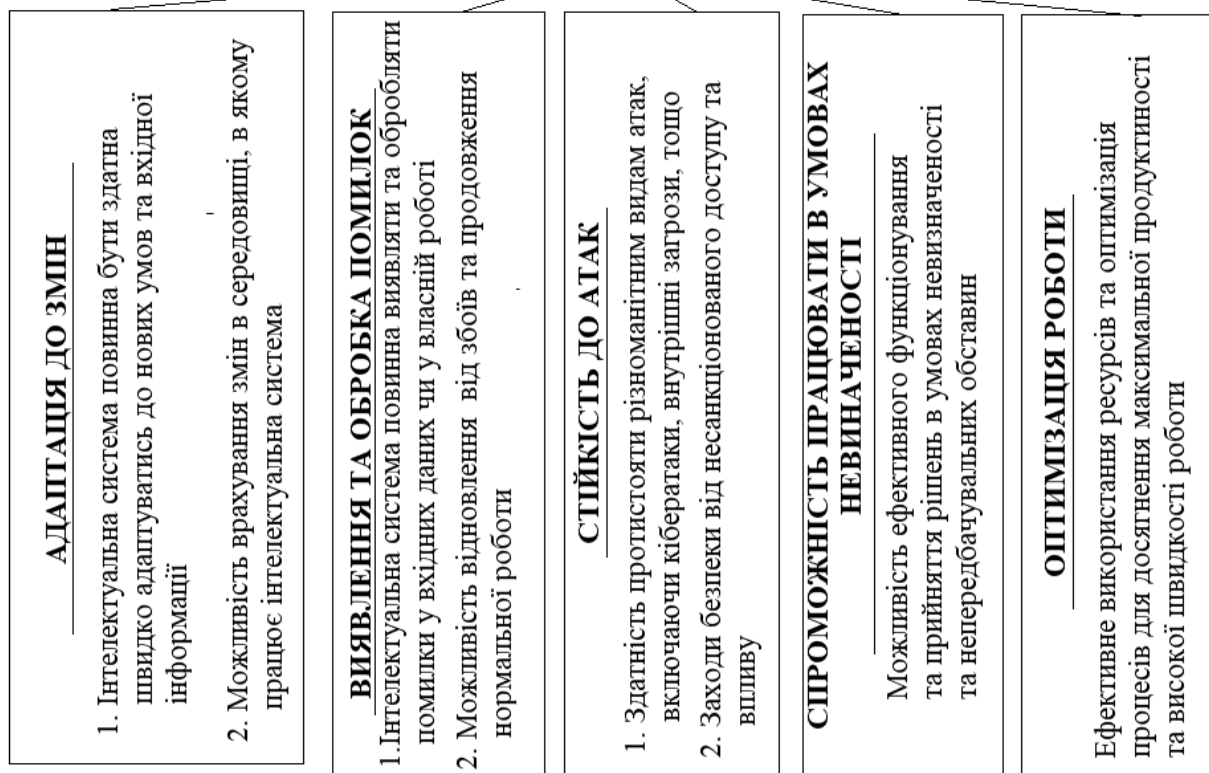


Рис. 2. Основні аспекти функціональної стійкості інтелектуальних систем під час ухвалення рішень

Наразі Україна потребує глобальної технологічної модернізації електромереж, в яких необхідно застосовувати регулятори напруги та регулятори потоків потужності. Це викликає необхідність розробки відповідного нормативно-правового та технічного забезпечення для інтеграції розосереджених джерел енергії, разом зі створенням концепції інтелектуальних систем автоматизації [17] і розумної мережі Smart Grid [4]. Система Smart Grid в загальному випадку складається з трьох основних рівнів: генерація енергії, передача та розподіл енергії та збут. На кожному з цих рівнів існують системи керування та контроль, які працюють в автоматичному режимі і об'єднані в єдину мережу, яка за допомогою ШІ забезпечує функціонування самої енергетичної системи вцілому.

Висновки

З широким застосуванням штучного інтелекту в енергетиці створюється «Розумна мережа», яка забезпечує автоматизоване функціонування всього енергетичного об'єкту, починаючи від генерації електричної енергії і закінчуючи її збутом. В результаті створена інтелектуальна система, замінюючи людину, виконує всі необхідні функції керування та контролю всього технологічного циклу. В цих умовах саму інтелектуальну систему необхідно постійно тестувати на спроможність виконання всіх необхідних функцій в реальному режимі часу за підтримки головних характеристик (параметрів) в межах робочого режиму. Основним шляхом досягнення цього є забезпечення функціональної стійкості самої інтелектуальної системи. В даній роботі було сформульовано

означення функціональної стійкості інтелектуальної системи під час ухвалення рішень та представлено її основні аспекти, виходячи з функцій штучного інтелекту, які при цьому застосовуються.

Література

1. <https://esu.com.ua/article-67277>
2. Суходоля О.М. Штучний інтелект в енергетиці: аналітична доповідь / Суходоля О.М. – К. : НІСД, 2022. – 49 с. – <https://doi.org/10.53679/NISS-analytrep.2022.09>.
3. Гончаров Є.В. Аналіз сучасних стратегій розвитку інформаційно-енергетичних систем / Є.В. Гончаров, І.В. Поляков, В.С. Марков, Н.В. Крюкова, Д.О. Бойков, М.К. Скрєбцов, І.В. Цєбрюк // Інтегровані технології та енергозбереження, 2020, №3, с. 75-83.
4. Шевчук О.А. Оптимізація енергоефективності економіки за допомогою технологічної концепції Smart Grid / О.А. Шевчук, Л.С. Борданова, Т.А. Наухацька // Економічний вісник НТУУ «КПІ», 2019, с. 400-414.
5. Іванченко Є.В. Аналіз поняття кіберстійкості критичної інфраструктури / Є.В. Іванченко, О.Г. Корченко, О.В. Зарицький, С.В. Зибін, Н.С. Вишевська // ЗАХИСТ ІНФОРМАЦІЇ, ТОМ 25, №4, ЖОВТЕНЬ-ГРУДЕНЬ 2023, с. 221-223.
6. Павлов В.В. Интеллектуальное управление: подходы. Результаты и перспективы развития. / В.В. Павлов, Ю.М. Шепетуха, С.В. Мельников, А.Е. Волков // Кибернетика и вычислительная техника, 2017, №1 (187), с. 30-48.
7. Буров Є. Вибір концептуалізації проблемної ситуації інтелектуальним агентом в задачах прийняття рішень / Є. Буров, І. Карпов // Information systems and Networks, 2023, Issue 13, p. 235-242.
8. Вишнівський В.В. Оцінка показників надійності інформаційних систем при обмеженій апріорній інформації / В.В. Вишнівський, Ю.В. Каргаполов, Ю.В. Березовська, М.Ю. Березівський, Р.В. Космінський // Sciences of Europe, 2021, №3, p. 8-14.
9. Заболотний С.В. Інформаційна технологія забезпечення функціональної стійкості систем моніторингу інформаційного простору в інтересах військ (сил) / С.В. Заболотний, В.О. Кацалан // Modern Information Technologies in the Sphere of Security and Defence, 2023, № 1(46), p. 141-144.
10. Сальник С.В. Аналіз методів прийняття рішень з забезпечення захищеності комунікаційних систем / С.В. Сальник, А.С. Сторчак, А.В. Мельник, А.С. Дівіцький // Наука і техніка Повітряних Сил Збройних Сил України, 2019, № 3(36), с. 122-131.
11. Вишнівський В.В. Статичні інформаційні показники якості інтелектуальних систем / В.В. Вишнівський, О.В. Зінченко, Ю.І. Катков, С.О. Серих // Телекомунікаційні та інформаційні технології, 2018, №2(59), с. 14-20.
12. Собчук В.В. Метод діагностування прихованих вимог в інформаційній системі на основі застосування дворівневої системи забезпечення функціональної стійкості / В.В. Собчук, М.О. Коваль, А.П. Мусієнко, О.Й. Мацько // Телекомунікаційні та інформаційні технології, 2019, №1(62), с. 22-30.
13. Ткаленко О.М. Інтелектуальні технології та системи штучного інтелекту для підтримки прийняття рішень / О.М. Ткаленко, А.О. Макаренко, О.В. Полоневич // Телекомунікаційні та інформаційні технології, 2019, №2(63), с. 53-59.
14. Собчук В.В. Основи забезпечення функціональної стійкості інформаційних систем підприємств в умовах впливу дестабілізуючих факторів / В.В. Собчук, О.В. Барабаш, А.П. Мусієнко / монографія. Київ: Міленіум, 2022. 272 с.
15. Собчук А.В. Метод оцінки функціональної стійкості бездротової сенсорної мережі / А.В. Собчук, О.В. Барабаш, А.П. Мусієнко // Телекомунікаційні та інформаційні технології, 2019, №3(64), с. 46-54.
16. Казначєєва А.В. Дослідження технології розумного очищення сонячних енергомодулів станції автономного живлення / А.В. Казначєєва, А.В. Заячковський, В.О. Завицький, К.П. Сторчак // ЗВ'ЯЗОК, 2024, №3, с. 43-47.
17. Аврунін О.Г. Інтелектуальні системи автоматизації: монографія / О.Г. Аврунін, С.І. Владов, М.В. Петченко, В.В. Семенець, В.В. Татарінов, Г.В. Тельнова, В.О. Філатов, Ю.М. Шмельов, Н.О. Шушляніна / Кременчук: Видавництво «НОВАБУК», 2021. – 322 с.
18. Трофімчук О.М. Інтелектуальні системи підтримки прийняття рішень у розвитку інфраструктури мегаполісу / О.М. Трофімчук, О.А. Стенін, А.О. Солдатова, І.Г. Дроздович // Системи дослідження та інформаційні технології, 2022, №2, с. 61-74.

19. Wen Y. On realization of Intelligent Decision Making in the Real World: A Foundation Decision Model Perspective / Ying Wen, Ziyu Wan, Ming Zhou, Shufang Hou, Zhe Cao, Chenyang Le, Jingxiao Chen, Zheng Tian, Weinan Zhang and Jun Wang // CAAI Artificial Intelligence Research, Vol. 2 Article № 9150026, 2023, p. 1-12. <https://doi.org/10.26599/AIR.2023.9150026>.
20. Rahman A. Intelligent Decision Support Systems – An Analysis of Machine Learning and Multicriteria Decision-Making Methods // App. Sci., 2023,13,12426. <https://doi.org/10.3390/app132212426>.
21. Poszler F. The impact of intelligent decision-support systems on humans`ethical decision-making: A systematic literature review and an integrated framework / Franziska Poszler, Benjamin Lange // Technological Forecasting & Social Change, 204 (2024), 123403, p. 1-19. <https://doi.org/10.1016/j.techfore.2024.123403>.

Одержано: 24.11.2024

Внутрішня рецензія отримана: 30.11.2024

Зовнішня рецензія отримана: 03.12.2024

Про авторів:

¹*Шуклін Герман Вікторович*,
кандидат технічних наук, доцент,
доцент кафедри інженерії програмного
забезпечення в енергетиці
e-mail: mathacadem-kiev@ukr.net.

²*Барабаш Олег Володимирович*,
доктор технічних наук, професор,
професор кафедри інженерії програмного
забезпечення в енергетиці
e-mail: bar64@ukr.net

³*Гребенніков Асаді Болдохоягович*,
заступник директора
e-mail: g_as_b@ukr.net

Місце роботи авторів:

^{1,2} Навчально-науковий інститут теплової
та атомної енергетики НТУ України
«КПІ імені Ігоря Сікорського».

³ Інститут програмних систем НАН України

ОПИС ПРОЦЕСУ ВЕБ-СЕРВІСУ. ТАБЛИЧНА ІНТЕРПРЕТАЦІЯ

Вирішення задач веб-сервісів на процесному рівні лишається актуальною проблемою вже протягом багатьох років. Дане дослідження спирається на попередні, де функціональна модель веб-сервісу та задачі цього рівня формалізуються онтологіями, побудованими на основі апарата дескриптивних логік. Але функціональна модель є статичною і не розглядає поведінки сервісу. Для аналогічної формалізації поведінки сервісу дескриптивна логіка має бути розширена темпоральними операторами, що і є фокусом даної роботи.

Для інтерпретації цих темпоральних операторів використовується табличне представлення. Дане дослідження розширює описану раніше табличну інтерпретацію темпоральних операторів майбутнього для LTL_{ACT} , використовуючи даний підхід для визначення табличного представлення операторів минулого часу LTL_{ACT} . В роботі наводиться приклад застосування табличної інтерпретації темпоральної дескриптивної логіки LTL_{ACT} до представлення динаміки конкретного прикладного процесу, а саме моделі адміністрування процесу подання на захист дисертаційної роботи на отримання ступеня доктора філософії.

Слід зауважити, що представлене в статті дослідження має певні обмеження, бо охоплює лише лінійний час і не підтримує розгалуження в часі, тобто не забезпечує можливість формалізації довільного процесу. Інтерпретація темпоральних операторів нелінійного часу є напрямком подальших досліджень.

Ключові слова: темпоральне розширення дескриптивної логіки, процесна модель веб-сервісу, таблична інтерпретація темпоральних операторів, онтології часу, часовий домен, перевірка моделі сервісу, опис динаміки веб-сервісу, лінійна темпоральна логіка, темпоральні оператори майбутнього, темпоральні оператори минулого, TDL онтологія сервісу, темпоральна структура.

V.A. Reznichenko, O.V. Zakharova

DESCRIPTION OF THE WEB-SERVICE PROCESS. TABULAR INTERPRETATION

Solving the tasks of web services at the process level is an actual problem for many years. This research is based on the previous ones, where the functional model of the web service and its tasks of this level are formalized by ontologies built on the basis of the descriptive logics. But the functional model is static and does not consider the behavior aspects of the web service. For a formalization of web service behavior, the descriptive logic should be extended by temporal operators. This is the focus of the research.

To interpret temporal operators a tabular representation is used. This study extends the tabular interpretation of future temporal operators for LTL_{ACT} previously described. It uses this approach to determine the tabular representation of LTL_{ACT} past tense operators. This paper gives an example of applying the tabular interpretation of LTL_{ACT} to describe the dynamics of a specific application process, namely, the model of the administration of the process of submitting a dissertation for the degree of Doctor of Philosophy.

It should be noted that the research presented in the article has certain limitations, because it covers only linear time and does not support branching in time. That is, it does not provide the possibility of formalizing an arbitrary process. The tabular interpretation of temporal operators of non-linear time is the direction of further researches.

Key words: temporal extension of descriptive logic, process model, Time ontology, tabular interpretation of temporal operators, time domain, checking model, defining the dynamics of the web service, linear temporal logic, temporal operators of Future, temporal operators of Past, TDL service ontology, temporal structure of the first order.

Вступ

Функціональна модель веб-сервісу розглядає сервіс як «чорну скриньку» з входами, виходами та стислим описом [1]. Тобто, користувач, обираючи сервіс(и), знає, що він

робить, які дані мають бути, за якими умовами цей сервіс виконується та результат його роботи, але не має уявлення, яким чином досягається цей результат. Однак не

завжди ефективно вирішення бізнес-задачі в сервісно-орієнтованій архітектурі можливе без урахування особливостей внутрішньої поведінки веб-сервісу.

Це особливо важливо, якщо сервіс є складним (не атомним), та, відповідно, реалізує процес, в середині якого є значущі події, стани, розгалуження, умови. Наприклад, може бути принциповим, щоб при бронюванні житла/квитків/товарів сервіс обов'язково перевіряв валідність платіжної картки замовника, а, може, й блокував потрібну суму коштів на ній і здійснював таку перевірку до завершення операції бронювання. Отже, це вже умова, яка має враховувати послідовність чи паралельність операцій в часі всередині процесу сервісу, іншими словами, часові характеристики операцій процесу. Слід зазначити, що потреба в описі часової інформації (у веб-системах, вцілому) породила різні онтології часу [2], які формалізують загальні часові поняття, такі як часові моменти, часові інтервали й календарні терміни тощо, і пропонують стандартизовані формати для представлення різних типів часової інформації. У даному дослідженні для визначення динаміки процесу будуть використовуватись поняття часових моментів, часових інтервалів та часових ланцюжків, які будуть формально описані нижче.

На процесному рівні веб-сервіс описується впорядкованим набором базових (знаючих) операцій та станів, які визначають його поведінку. Семантизація даних та поведінки веб-сервісу на цьому рівні полягає у збагаченні специфікації сервісу семантичними анотаціями, які пов'язують опис сервісу із прикладним доменом. Зазвичай це посилання на онтологію прикладної сфери, де онтологія є загальноприйнятною формалізацією домену. Якщо за формальну мову представлення онтології (статичних сутностей) обрати дескриптивну логіку (DL), то для визначення динаміки процесу, вона має бути розширена темпоральними операторами (TDL).

Слід зазначити, що доцільність використання DL онтологій у вирішенні задач веб-сервісів було обґрунтовано в [3].

Темпоральні дескриптивні логіки та бази знань. Основні поняття та визначення

Спочатку наведемо стислий опис базових понять DL компонента [4], а потім його темпоральне розширення. Всі визначення будуть базуватися на дескриптивній логіці *ALC*, що використовується просто як прототип.

Визначення 1. (Синтаксис) Нехай N_C , N_R , N_I не порожні множини імен, відповідно, концептів, ролей та індивідів, що попарно не перетинаються. Базовий концепт має форму $A \in N_C$, $\neg A$ – доповнення концепту A (інверсія). Аксиоми включення концептів: $B \sqsubseteq C$, $B, C \in N_C$. Твердження мають форму $A(a)$ або $P(a, b)$, де $A \in N_C$, $P \in N_R$, $a, b \in N_I$.

ТВох дескриптивної логіки є кінцева множина включень концептів, тоді як АВох – кінцева множина тверджень індивідів.

Семантика дескриптивної логіки вводиться через поняття інтерпретації.

Визначення 2. (Семантика) Інтерпретацією є пара $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$, де $\Delta^{\mathcal{J}}$ – непорожня множина, що називається доменом, та \mathcal{J} – функція інтерпретації, яка призначає кожному $A \in N_C$ множини $A^{\mathcal{J}} \subseteq \Delta^{\mathcal{J}}$, кожному $P \in N_R$ бінарне відношення $P^{\mathcal{J}} \subseteq \Delta^{\mathcal{J}} \times \Delta^{\mathcal{J}}$, та кожному $a \in N_I$ елемент $a^{\mathcal{J}} \in \Delta^{\mathcal{J}}$.

\mathcal{J} є моделлю $B \sqsubseteq C$, якщо $B^{\mathcal{J}} \subseteq C^{\mathcal{J}}$, моделлю $A(a)$, якщо $a^{\mathcal{J}} \in A^{\mathcal{J}}$, моделлю $P(a, b)$, якщо $(a^{\mathcal{J}}, b^{\mathcal{J}}) \in P^{\mathcal{J}}$. Якщо \mathcal{J} є моделлю всіх включень концептів з ТВох \mathcal{T} , будемо позначати це як $\mathcal{J} \models \mathcal{T}$. Якщо \mathcal{J} є моделлю всіх тверджень індивідів з A , то позначаємо це як $\mathcal{J} \models A$.

Вважатимемо, що всі інтерпретації \mathcal{J} задовольняють припущенню унікальності імен (UNA), тобто для всіх $a, b \in N_I$, якщо $a \neq b$, то $a^{\mathcal{J}} \neq b^{\mathcal{J}}$.

База знань є сукупністю всіх тверджень про предметну сферу, тобто є поєднанням аксіом TBox та тверджень ABox.

Це загальні визначення синтаксису та семантики дескриптивних логік. Існує ціла сім'я дескриптивних логік, які відрізняються складом елементів (концептів, ролей, індивідів), а також множиною конструкторів та аксіом, що підтримуються та визначаються на цих елементах. Дане дослідження спирається на базову дескриптивну логіку *ALC*.

Темпоральні дескриптивні логіки уможливають керування часовими даними в DL онтологіях, що фіксують часові обмеження на інтенціональному рівні.

Визначення 3. Часовий домен [5], у загальному випадку, визначається парою $(T, <)$, де T – непорожня множина моментів часу, а $<$ – лінійне відношення порядку на T .

Більшість загальноприйнятих структур, що відповідають цьому визначенню, базуються на числових множинах. Наприклад, множині натуральних $(N, <)$, цілих $(Z, <)$ чи дійсних $(R, <)$ чисел, де відношення впорядкування зазвичай інтерпретується як «менше, ніж». З моментів часу можуть будуватися більш складні часові елементи (інтервали, ланцюжки тощо).

Визначення 4. Темпоральна інтерпретація DL (TDL) $\mathcal{J} = (\Delta, \cdot^{\mathcal{J}})$ складається з непорожнього домена Δ та функції інтерпретації $\cdot^{\mathcal{J}}$, яка відображає:

- кожне ім'я індивіда $a \in N_I$ в елемент $a^{\mathcal{J}} \in \Delta$;
- кожне ім'я концепта $A \in N_C$ в підмножину $A^{\mathcal{J}} \subseteq T \times \Delta$;
- кожне ім'я ролі $R \in N_R$ в підмножину $R^{\mathcal{J}} \subseteq T \times \Delta \times \Delta$.

У цьому визначенні T – домен, що представляє час $(T, <)$. Тобто, концепти та ролі (їхній склад) змінюються в часі. У даній темпоральній інтерпретації імена індивідів НЕ змінюються в часі й інтерпретуються од-

наково на всій часовій осі, тобто є фіксованими.

Визначена інтерпретація – спеціальний випадок темпоральної структури першого порядку, не містить функціональних символів, предиката рівності та n -місних предикатів з $n > 2$.

Зазвичай темпоральна база знань містить глобальний (незмінний в часі) TBox та сенсорні дані (ABox-и) для визначеної множини моментів або інтервалів часу. На сьогодні відомо чимало темпоральних логік, які, перш за все, відрізняються способом визначення часової складової: лінійна (LTL), інтервальна (ITL), зрізана лінійна темпоральна логіка (TLTL) тощо. Для специфікації конкретної темпоральної логіки важливо, чи є час безперервним або дискретним, лінійним або розгалуженим, як визначається майбутнє й минуле, що таке поточний час, структура (моменти, інтервали, ланцюжки інтервалів) та способи вимірювання часу.

Як вже було зазначено, в даному дослідженні будемо розглядати дискретний час (лінійно впорядкована цілочисленна часова вісь) та спиратися на множину темпоральних операторів логіки LTL, хоча вона є недостатньою для опису процесу сервісу у загальному випадку, адже не підтримує розгалуження у часі, що, зазвичай, можуть бути притаманними процесу. Інтерпретація темпоральних операторів STL*, яка узагальнює LTL та STL й дозволяє описувати розгалуження процесу у часі, є предметом подальших досліджень.

Метою є розширення дескриптивної логіки темпоральними операторами стандартної темпоральної логіки та аналіз отриманої моделі в аспекті застосування для опису лінійного процесу веб-сервісу та умов його поведінки. Тобто даний підхід припускає використання темпоральних операторів у базовій дескриптивній логіці як конструкторів концептів, що уможливають опис динаміки сервісу через зміну членства індивідів у концептах DL, ролей DL та ABox, вцілому в часі. Вважатимемо, що TBox лишається незмінним.

Вбудовування темпоральних операторів в DL, зокрема DL \mathcal{ALC} , здійснюється наступним чином (синтаксис $LTL_{\mathcal{ALC}}$) [6]:

$T \mid \perp \mid A \mid \neg C \mid C \wedge D \mid C \vee D \mid \exists R.C \mid \forall R.C$ – конструктори концептів \mathcal{ALC} ,

$\circ C \mid \diamond C \mid \square C \mid C \mathcal{U} D \mid C \mathcal{R} D$ – оператори LTL майбутнього,

$\sigma^{-1}C \mid \diamond^{-1}C \mid \square^{-1}C \mid C \mathcal{U}^{-1}D \mid C \mathcal{R}^{-1}D$ – оператори LTL минулого, де C та D довільні концепти.

Семантики наведених темпоральних операторів визначаються як:

$(\circ C)^{\mathcal{J}} = \{(n, d) \mid (n+1, d) \in C^{\mathcal{J}}\}$ – наступний; (1)

$(\diamond C)^{\mathcal{J}} = \{(n, d) \mid \exists m \geq n (m, d) \in C^{\mathcal{J}}\}$ – має місце в певний момент часу в майбутньому; (2)

$(\square C)^{\mathcal{J}} = \{(n, d) \mid \forall m \geq n (m, d) \in C^{\mathcal{J}}\}$ – має місце завжди в майбутньому; (3)

$(C \mathcal{U} D)^{\mathcal{J}} = \{(n, d) \mid \exists m \geq n ((m, d) \in D^{\mathcal{J}} \wedge (k, d) \in C^{\mathcal{J}} \text{ для } n \leq k < m)\}$ – має місце доки; (4)

$(C \mathcal{R} D)^{\mathcal{J}} = \{(n, d) \mid \text{або } \forall m \geq n (m, d) \in D^{\mathcal{J}}, \text{ або } \exists k \geq n ((k, d) \in C^{\mathcal{J}} \wedge (m, d) \in D^{\mathcal{J}} \text{ для } n \leq m \leq k)\}$ – звільнити (Release). (5)

Оператори минулого часу є інверсними відносно аналогічних операторів майбутнього та мають наступну семантику:

$(\sigma^{-1}C)^{\mathcal{J}} = \{(n, d) \mid n > 0 \wedge (n-1, d) \in C^{\mathcal{J}}\}$ – попередній; (6)

$(\diamond^{-1}C)^{\mathcal{J}} = \{(n, d) \mid \exists m \leq n (m, d) \in C^{\mathcal{J}}\}$ – має місце в певний момент часу в минулому; (7)

$(\square^{-1}C)^{\mathcal{J}} = \{(n, d) \mid \forall m \leq n (m, d) \in C^{\mathcal{J}}\}$ – має місце завжди в минулому; (8)

$(C \mathcal{U}^{-1} D)^{\mathcal{J}} = \{(n, d) \mid \exists m \leq n ((m, d) \in D^{\mathcal{J}} \wedge (k, d) \in C^{\mathcal{J}} \text{ для } m < k \leq n)\}$ – має місце поки (Until) в минулому, або з того моменту (Since); (9)

$(C \mathcal{R}^{-1} D)^{\mathcal{J}} = \{(n, d) \mid \text{або } \forall m \ 0 \leq m \leq n (m, d) \in D^{\mathcal{J}}, \text{ або } \exists k \leq n ((k, d) \in C^{\mathcal{J}} \wedge (m, d) \in D^{\mathcal{J}} \text{ для } n \geq m \geq k)\}$ – звільнити (Release) в минулому, або тригер (Trigger). (10)

Наведені постулати визначають темпоральну інтерпретацію DL.

Табличне представлення ТДЛ

В [7] автори запропонували табличну інтерпретацію майбутнього часу темпоральної DL. Цінність табличного представлення полягає в його простоті, наочності, а також можливості легкого конвертування у таблиці для зберігання відповідних даних та знань і подальшої роботи з ними. Ідея полягає в тому, що семантика всіх операторів подається у вигляді таблиць, де по вертикалі розташовуються індивіди концепта, а по горизонталі – моменти часу.

На часовій осі визначаються три типи елементів часу (що розглядаються в межах даного дослідження):

- момент часу – одна комірка таблиці;

- інтервал часу – поспіль розташовані по горизонталі комірки. У виродженому випадку інтервал може складатися з однієї комірки, тобто початок та кінець інтервалу співпадають;

- часовий ланцюжок – впорядкована послідовність інтервалів, що розташовані по горизонталі. Інтервали у ланцюжку не можуть ані перетинатися, ані торкатися. У виродженому випадку ланцюжок може містити лише один інтервал. Ланцюжок, який складається з двох (чи більше) комірок, НЕ може розглядатися як два (чи більше) самостійних моментів часу.

В табличній інтерпретації кожний темпоральний концепт (атомарний чи складовий) є двомірною таблицею, один вимір якої є вісь часу, другий – індивіди концепта. Аналогічно для ролі. З точки зору поняття темпоральної інтерпретації та застосування темпоральних операторів роль не відрізняється від концепта. Різниця полягає лише в тому, що для концепта ці поняття визначаються для множини індивідів, а для ролі – для множини пар індивідів. Тобто, роль представляється двомірною таблицею, один

Сенс оператора $\diamond C$ - «у майбутньому настане такий момент часу, коли буде мати місце концепт C » (у кожному рядку знаходиться перша права сіра комірка і всі комірки ліворуч аж до першої, перефарбовуються на сірі).

C	■																		
$\diamond C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C																			■
$\diamond C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■				■												■
$\diamond C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■																■
$\diamond C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Темпоральний оператор $\square C$ означає, що «з даного моменту завжди в майбутньому буде мати місце концепт C » (для кожного рядка таблиці інтерпретації концепта розміщена перша права біла комірка, і всі комірки ліворуч фарбуються в білий колір).

C	■																		
$\square C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C																			■
$\square C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■				■												■
$\square C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■																■
$\square C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Оператор $C \mathcal{U} D$ означає, що «відносно даного моменту часу настане такий момент у майбутньому, що в ньому буде мати місце концепт D , й протягом усього цього діапазону (тобто доки - UNTIL), буде мати місце концепт C ». Іншими словами, у майбутньому D стане істинним, а до того моменту буде істинним C . Для кожного безперервного інтервалу сірих комірок таблиці інтерпретації концепту C розміщена крайня права сіра комірка D (в межах часового інтервалу даного діапазону C) і цей діапазон обрізається праворуч

відносно цієї сірої комірки D . Якщо в межах інтервалу сірих C немає жодної сірої D , то цей діапазон не зараховується до результату оператора \mathcal{U} .

C			■																
D																			
$C \mathcal{U} D$			■																

Більш детальний розгляд TDL операторів майбутнього та їхня таблична інтерпретація наведена в [7]. Далі аналогічним чином визначимо табличну інтерпретацію для темпоральних операторів минулого часу та введемо кілька нових операторів, які будуть виражатися через базові і таким чином не підвищуватимуть виразну спроможність TDL, але забезпечать зручніші формальні засоби у вирішенні прикладної задачі.

Таблична інтерпретація темпоральних операторів минулого. Семантика темпоральних операторів минулого часу була визначена вище, в постулатах (6) – (10). Та, як було зазначено, оператори минулого часу є інверсними щодо аналогічних операторів майбутнього.

Так оператор O^{-1} є інверсним до оператора майбутнього O . $O^{-1}C$ означає, що «у попередній момент часу мав місце концепт C ». Іншими словами, концепт $O^{-1}C$ є істинним в певний момент часу n , якщо C є істинним у момент часу $n-1$. В табличній інтерпретації це означає зсув сірих комірок на одну праворуч. Продемонструємо дію оператора на прикладі.

C			■																
$O^{-1}C$	■																		
C																			■
$O^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■				■												■
$O^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■																■
$O^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Оператор \diamond^{-1} є інверсним до темпорального оператора майбутнього \diamond . Сенс оператора $\diamond^{-1}C$ - «в минулому існував такий момент часу, коли мав місце концепт C ». Іншими словами, концепт $\diamond^{-1}C$ є істинним на даний момент часу, якщо C був істинним в певний момент часу в минулому. В табличній інтерпретації для кожного рядка таблиці інтерпретації концепта розташована перша ліва сіра комірка і всі наступні комірки аж до останньої перетворюються на сірі. Якщо такої комірки не існує, то весь рядок залишається білим. Таблиця нижче демонструє дію оператора на прикладі.

C	■																		
$\diamond^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C																			■
$\diamond^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■			■	■												
$\diamond^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■															■	■
$\diamond^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Оператор \square^{-1} є інверсним до темпорального оператора \square . $\square^{-1}C$ означає, що «з даного моменту часу завжди в минулому мав місце концепт C ». Іншими словами, концепт $\square^{-1}C$ є істинним в даний момент, якщо C є істинним у всі моменти часу в минулому (відносно даного моменту). В табличній інтерпретації для кожного рядка таблиці інтерпретації концепта розташована перша ліва біла комірка, і всі наступні комірки аж до останньої перетворюються на білі. Якщо такої комірки не існує, то весь рядок лишається сірим.

C	■																		
$\square^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C																			■
$\square^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C			■			■	■												
$\square^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
C		■																■	■
$\square^{-1}C$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Сенс темпорального оператора минулого $C\mathcal{U}^{-1}D$ - «щодо даного моменту часу в минулому існував такий момент часу, в якому мав місце концепт D , і з того моменту (SINCE) мав місце концепт C ». Іншими словами, колись у минулому мав місце концепт D , і відтоді був істинним концепт C . У табличній інтерпретації в результат дії оператора потрапляють ті сірі комірки C , що отримуються наступним чином: для кожного безперервного інтервалу сірих C розташована крайня ліва сіра комірка D (в межах цього часового інтервалу C) і по ній обрізається даний сірий інтервал ліворуч. Якщо в межах діапазону сірих C немає жодної сірої D , то цей інтервал не включається до результату \mathcal{U}^{-1} . Продемонструємо дію оператора на прикладі.

C			■	■	■	■													■	■
D							■	■											■	
$C\mathcal{U}^{-1}D$	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Оператор $C\mathcal{R}^{-1}D$ означає: «до результату потрапляють всі інтервали/ комірки концепта C та додатково, коли відносно даного моменту часу був такий момент часу в минулому, що в ньому мав місце C , і протягом всього цього інтервалу мав місце концепт D , а також ланцюжок, де був істинним $\square^{-1}D$ ». Таблична інтерпретація оператора демонструється на прикладі нижче. До результату дії оператора потрапляють:

видалення відповідної кількості інтервалів праворуч.

S_1^L – вибір крайнього лівого інтервалу в ланцюжку. Можна сказати, що дана операція за дією є протилежною до $D_1^L C$, тому що в результаті в таблиці інтерпретації залишається тільки крайній лівий інтервал. $S_1^L C$ обчислюється за формулою: $S_1^L C = C \sqcap \square^{-1}(C \sqcup \diamond^{-1} C)$.

Продемонструємо дію оператора на прикладі інтерпретації концепта C , що визначена в таблиці нижче.

C																				
$\diamond^{-1} C$																				
$\neg \diamond^{-1} C$																				
$C \sqcup \neg \diamond^{-1} C$																				
$\square^{-1}(C \sqcup \neg \diamond^{-1} C)$																				
$C \sqcap \square^{-1}(C \sqcup \neg \diamond^{-1} C)$																				

S_2^L – вибір другого зліва інтервалу у ланцюжку. Результат досягається наступною послідовністю операцій: видаляється крайній лівий інтервал ланцюжка, потім обирається крайній лівий інтервал в тому ланцюжку, що лишився. Обчислюється за формулою: $S_2^L C = D_1^L C \sqcap \square^{-1}(D_1^L C \sqcup \diamond^{-1} D_1^L C)$.

S_n^L - вибір n -го (третього і далі) інтервалу ліворуч у ланцюжку здійснюється так само.

S_n^R - вибір n -го (першого, другого і т.д.) інтервалу праворуч у ланцюжку, виконується симетрично аналогічному вибору ліворуч, описаного вище.

DI_1^L – видалення першої лівої комірки в інтервалі. Результат досягається наступною послідовністю раніше визначених операцій: $DI_1^L C = C - (C - O - 1C)$. Продемонструємо дію даного оператора на прикладі:

C																				
$O^{-1} C$																				
$C - O^{-1} C$																				
$C - (C - O^{-1} C)$																				

Видалення двох чи більше комірок в інтервалі здійснюється ітераційним виконанням оператора DI_1^L , наприклад, $DI_2^L C = DI_1^L(DI_1^L C)$ – видалення двох крайніх лівих комірок в інтервалі.

DI_1^R – видалення першої правої комірки в інтервалі. Даний оператор є симетричним до попереднього та обчислюється наступним чином: $DI_1^R C = C - (C - OC)$. Таблиця нижче демонструє побудову та дію оператора на прикладі.

C																				
OC																				
$C - OC$																				
$C - (C - OC)$																				

Видалення двох чи більше комірок в інтервалі здійснюється ітераційним виконанням $DI_1^R C$, так, $DI_2^R C = DI_1^R(DI_1^R C)$ – видалення двох крайніх правих комірок в інтервалі і так далі.

AI_1^L – додавання до інтервалу однієї комірки ліворуч. Результат обчислюється за формулою: $AI_1^L C = C \sqcup OC$. Продемонструємо роботу даного оператора на прикладі в таблиці нижче.

C																				
OC																				
$C \sqcup OC$																				

Якщо інтервал примикає до лівого боку таблиці, то до нього нічого не додається.

Так само можна додати ліворуч будь-яку кількість комірок.

Оператор AI_1^R додає до інтервалу одну комірку праворуч, є симетричним попередньому оператору та обчислюється за формулою: $AI_1^R C = C \sqcup O^{-1} C$. Наведемо приклад, що демонструє роботу оператора, в таблиці нижче.

та її застосування для представлення динаміки процесної моделі веб-сервісу.

Визначення веб-сервісу, пошукового запиту та задач виявлення та композиції веб-сервісів на функціональному рівні засобами дескриптивних логік уже було наведено у попередніх дослідженнях. Але функціональна модель сервісу не розкриває його поведінки, а дескриптивна логіка сама собою не дає можливості опису динаміки. Саме використання темпоральних операторів забезпечує формальний інструментарій для опису процесу.

Тому основним питанням даного дослідження є розширення дескриптивної логіки темпоральними операторами та інтерпретація їх через табличне представлення. Слід зазначити, що табличну інтерпретацію для операторів майбутнього часу лінійної темпоральної логіки було наведено в попередніх дослідженнях. Дана робота є розширенням, що, за аналогією, вводить табличну інтерпретацію темпоральних операторів минулого часу LTL ALC.

У роботі наводиться приклад опису прикладного процесу веб-сервісу засобами табличної інтерпретації LTL, розширеної темпоральними операторами. Але для опису довільної поведінкової моделі сервісу не достатньо засобів LTL ALC, тому що вона не охоплює операторів розгалуження. Вивчення та визначення табличної інтерпретації темпоральних операторів для нелінійних темпоральних логік, зокрема CTL, є напрямком подальших досліджень.

Література

1. Web Service composition: Semantic Links based approach. Freddy L'ecu', Doctor of Philosophy, 2008.
2. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. ACM Trans. Asian Lang. Inf. Process. 3(1) (2004) 66-85.
3. Захарова О. Використання дескриптивних логік в проблематиці web-сервісів. Проблеми програмування. 2015. N 1. С. 38–50.
4. Gutiérrez-Basulto, V., Klarman, S.: Towards a unifying approach to representing and querying temporal data in description logics. In: Krötzsch, M., Straccia, U. (eds.) RR 2012. LNCS, vol. 7497, pp. 90–105. Springer, Heidelberg (2012).

5. Montanari, A., Chomicki, J.: Time domain. In: Encyclopedia of Database Systems. (2009) 3103-3107.
6. Carsten Lutz, Frank Wolter, and Michael Zakharyashev. Temporal description logics: A survey. In 15th International Symposium on Temporal Representation and Reasoning, 2008. IEEE Computer Society Press, P. 3–14.
7. Reznichenko V., Chystiakova I. Table interpretation of the temporal description logic LTLALC *Prombles in programming 2022; 3-4: 216-230*
8. Zakharova O. Defining and resolving Web-services discovery problems using description logics formalism. Problems in programming. 2017. N 4. P. 66–78.
9. Schlingloff H., Martens A., Schmidt K. Modeling and Model Checking Web Services. Electronic Notes in Theoretical Computer Science. (2005) V. 126. P. 3-26.

References

1. Web Service composition: Semantic Links based approach. Freddy L'ecu', Doctor of Philosophy, 2008.
2. Hobbs, J.R., Pan, F.: An ontology of time for the semantic web. ACM Trans. Asian Lang. Inf. Process. 3(1) (2004) 66-85.
3. Zakharova O. Descriptive logic using in Web-service problems. Problems in programming. 2015. N 1. С. 38–50.
4. Gutiérrez-Basulto, V., Klarman, S.: Towards a unifying approach to representing and querying temporal data in description logics. In: Krötzsch, M., Straccia, U. (eds.) RR 2012. LNCS, vol. 7497, pp. 90–105. Springer, Heidelberg (2012).
5. Montanari, A., Chomicki, J.: Time domain. In: Encyclopedia of Database Systems. (2009) 3103-3107.
6. Carsten Lutz, Frank Wolter, and Michael Zakharyashev. Temporal description logics: A survey. In 15th International Symposium on Temporal Representation and Reasoning, 2008. IEEE Computer Society Press, P. 3–14.
7. Reznichenko V., Chystiakova I. Table interpretation of the temporal description logic LTLALC *Prombles in programming 2022; 3-4: 216-230*
8. Zakharova O. Defining and resolving Web-services discovery problems using description

logics formalism. Problems in programming. 2017. N 4. P. 66–78.

9. Schlingloff H., Martens A., Schmidt K. Modeling and Model Checking Web Services. Electronic Notes in Theoretical Computer Science. (2005) V. 126. P. 3-26.

Одержано: 06.11.2024

Внутрішня рецензія отримана: 11.11.2024

Зовнішня рецензія отримана: 12.11.2024

Про авторів:

Резніченко Валерій Анатолієвич,
кандидат фізико-математичних наук,
провідний науковий співробітник.
<http://orcid.org/0000-0002-4451-8931>

Захарова Ольга Вікторівна,
кандидат технічних наук,
старший науковий співробітник.
<http://orcid.org/0000-0002-9579-2973>.

Місце роботи авторів:

Інститут програмних систем
НАН України,
E-mail: ozakharova68@gmail.

N. A. Sydorov

ECOSYSTEM RESEARCH IN SOFTWARE ENGINEERING

The application of the concept of the ecosystem in the software engineering shows the existence of the same problems regarding the definition of the concept of the ecosystem and its use for research that still exists in ecology. Justification for applying the concept of the ecosystem in an area that differs significantly from the ecology area, as in our case, requires researchers to look for analogies. This primarily applies to landscape, energy and matter transfer chains (trophic chains) and nutritional cycles. Until such analogies will be found in software engineering, the ecosystem research is nothing more than system analysis, and the concept of the ecosystem is an attractive concept. The purpose of this position article is to draw the attention of the software engineering community to ecosystem research. Three concepts of ecosystems in ecology, software and software engineering are considered. The composition and essence of ecosystem research in the software engineering are given. The literature review of on the state of ecosystem research in the software was carried out.

Keywords: Software engineering, software ecosystem, software engineering ecosystem, software engineering ecosystem cycles.

M.O. Сидоров

ЕКОСИСТЕМНІ ДОСЛІДЖЕННЯ В ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Застосування концепції екосистеми в інженерії програмного забезпечення показує існування тих самих проблем щодо визначення концепції екосистеми та її використання для досліджень, які все ще існують в екології. Обґрунтування застосування поняття екосистеми на території, яка суттєво відрізняється від території екології, як у нашому випадку, вимагає від дослідників пошуку аналогій. В першу чергу це стосується ландшафту, ланцюгів переносу енергії та речовини (трофічних ланцюгів) і циклів харчування. Поки такі аналогії не будуть знайдені в інженерії програмного забезпечення, дослідження екосистеми є не чим іншим, як системним аналізом, а концепція екосистеми є привабливою концепцією. Мета цієї позиційної статті — привернути увагу спільноти інженерії програмного забезпечення до дослідження екосистем. Розглянуто три концепції екосистем, а саме в екології, програмному забезпеченні та інженерії програмного забезпечення. Наведено склад та сутність екосистемних досліджень у інженерії програмного забезпечення. Проведено огляд літератури щодо стану екосистемних досліджень у інженерії програмного забезпечення.

Ключові слова: інженерія програмного забезпечення, екосистема програмного забезпечення, екосистема інженерії програмного забезпечення, екосистемні цикли інженерії програмного забезпечення

Introduction

The application of the concept of the ecosystem in the software engineering shows the existence of the same problems regarding the definition of the concept and its use for research that still exists in ecology [1-3]. Justification for applying the concept of an ecosystem in an area that differs significantly from ecology, as in our case, requires researchers to look for analogies. This primarily applies to landscape, the energy and matter transfer chains (trophic chains) and nutritional cycles. Until similar constructions are found in the software engineering, ecosystem research will

be nothing more than system analysis, and the ecosystem will be an attractive concept but without sense.

The article [4] showed the expediency of applying the concept of the ecosystem in the software while preserving essence of the biology ecosystem. In the article [5], the concept of the software engineering ecosystem introduced. The purpose of this article is to draw the attention of software engineering community to ecosystem research. To justify position, the types of ecosystems - the biology ecosystem, software ecosystem, software engineering eco-

system and features of the ecosystem research in software engineering are considered. The state of the ecosystem research in the software by the review of the relevant literature is presented.

To the using of the ecosystem concept

Consider three using of concept of ecosystem - the ecology ecosystem, the software ecosystem and the ecosystem of software engineering. Epistemology of the ecosystem concept can be found in article [1].

Ecology ecosystem. The term "ecosystem" should denote a group of organisms living in a certain area, interacting with their physical environment in such a way that the flow of energy leads to a well-defined trophic structure, biotic diversity and material cycles [6]. The concept of the ecosystem can be applied to various ecological objects. This application is justified by the fact that they demonstrate similar forms of relationships (energy and material cycles). The concept of the ecosystem is based on the concept of the system, but the ecosystem research differs from system research. This difference is expressed is that living components are necessarily present in the ecosystem and demands are made requirements that include study of internal cycles of matter, energy flows, and nutrient cycle, and due to the presence of non-living components, the ecosystem is characterized interdisciplinarity [6]. It is clear that the application of the concept in software engineering should be guided by the same requirements and a necessary condition for their satisfaction should be the presence of analogues of interactions specified for ecological ecosystems.

Software ecosystem. This a widespread term today that has many interpretations. In the review article [7], four definitions of software ecosystems are given. Analyzing these definitions, the authors identify three main elements that combine these definitions. This is platform, business and relationships. In addition, a generalized definition of the software ecosystem as the interaction of a set of subjects on top of a general technological platform, leading to a number of solutions in software or services is formulated. The article [8] has no new terms, but definitions are given that

are related to the software ecosystem: mobile learning ecosystems/mobile ecosystem, open source free software ecosystem, open ecosystem, digital (business) ecosystem. Here, these definitions stand next to the definition of the software ecosystem. As for cycles and chains, the work [7] indicates that the software ecosystem is an association that manages the ecosystem, executing the platform, creating and applying rules, processes, business procedures, setting and controlling quality standards and/or organizing relationships with the actor.

Software engineering ecosystems. In the work [5], based on the assumption that in software engineering, as well as in ecology, there should be a wider range of ecosystems than software ecosystems known in the literature, the concept of the software engineering ecosystems was introduced. The classification of ecosystems and the concepts of flows and chains similar to biological ecosystems was proposed. As in [5], in this article we will keep to the following point of view.

The ecosystem is a concept, not a real entity and is based on the concept of the system. The concept can be applied to biogeocenosis on the landscape that has system boundary, within the software engineering territory [5]. Each system, in the defined landscape of the software engineering territory that is open to material and energy flows and their analogies, and contains at least two living beings (organisms) and meet the requirements of the ecosystems, can studied as a software engineering ecosystem. However, since the researcher determines the boundary of the landscape based on the goals of the study that the previously determined boundary is not always the system boundary and biogeocenosis on the landscape is not will be a system. Further will be studied only system biogeocenosis. As an analogue of energy transferred in biological ecosystems in trophic cycles, it is proposed to use the concept of value based on the energy theory of value [9]. To determine ecosystems of the software engineering territory, we will use a structural ecosystem approach [10]. Therefore, in the ecosystem, in addition to the abiotic component and value, we consider four more types of elements of the ecosystem structure [10]. Namely, firstly, the types of activities that determine discrete actions that must be

carried out in order for the value proposal to materialize. Secondly, actors who are subjects and who carry out activities. Thirdly, positions that determine where subjects are in the flow of activity and characterize who transfers value to whom; fourthly, connections that determine transfers value and artifacts between subjects. There are activities at the center of the definition of the ecosystem. To make the ecosystem sustainable, actors need to interact in such a way that the value materializes. Therefore, as an analogue of the trophic chain, we will use the chain of interdependent value creation by actors performing activities in the landscape using and producing the abiotic components (artifacts). The activities of the actors play an important role in the ecosystem, defining its functions and services. Sometimes one actor can own the platform (a product, service or technology) and plays the leader role in the software engineering ecosystem then the ecosystem is called the platform ecosystem [11].

In the same way, we will consider software landscapes on which artificial biota can operate - computer programs, as well as landscapes on which both types of biota (human and computer programs) act [12]. For a system formed by artificial biota, a chain of formation of the emergence function of the system can be specified as an interdependent value creation chain.

The place the software engineering ecosystems in the living systems

The software engineering will be considered as part of human activity aimed at transforming the biosphere into an anthroposphere (noosphere). Transforming is carried out by human influence on the biosphere by his activity and thought. Although thought is not a form of energy, nevertheless under its influence, processes occur and artifacts are created that change the biosphere. Occupying a certain area of the biosphere, software engineering carries out a variety of activities aimed at solving practical tasks often occurring in other territories, thereby playing a special role in the context of the anthroposphere. In general, the following types of the software engineering activity on human activity habitats can be distinguished (Fig. 1):

- software engineering education and research. The habitat consists of universities and fundamental research organizations;
- software engineering. The habitat is organizations that transform the results of the fundamental research to recommendations for practical application;
- software development and maintenance. The habitat consists of organizations that create and maintain software products for different domains;
- software market. The habitat includes organizations supplying software products;
- software operation. The habitat is defined on other territories on that the software product is used;
- executing software. The habitat is the software products themselves.

It is obvious that these types of activities, although specific, are carried out in the context of the biosphere and, therefore, the biogeochemical cycles and the waste activities are inherent the territory of software engineering and they must be studied in the software engineering ecosystems (Fig. 1).

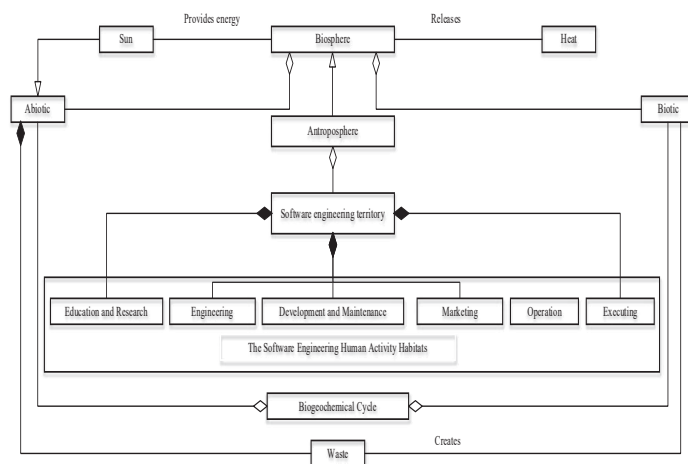


Fig. 1. The types of the software engineering activity on human activity habitats

Considering the territory of software engineering as part of the living system in which take place biogeocenosis, can apply to them the concept of the ecosystem to it and after that will study by performing the ecosystem research. Since the concept of a software engineering ecosystem is applied in the context of biological ecosystems that the relevant parts of both biotic and abiotic components, as well as biogeochemical cycles taking place

in the territory of software engineering, may be of interest also. However, new aspects of software engineering ecosystem research should be related with locating of the territory in the anthroposphere. Here the anthropos are new biotic components and the anthropogenic artifacts are new abiotic components, and arise the new networks and cycles inherent only in the software engineering ecosystems (Fig. 2). Therefore, along with the known ones (from ecology), there should be new directions of the software engineering ecosystem research.

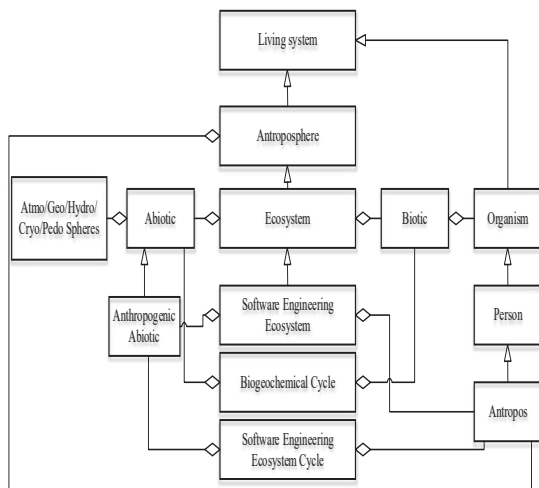


Fig. 2. Software engineering ecosystems in the anthroposphere

Thus, in living systems of anthroposphere, the object of the ecosystem research in the territory of the software engineering will be the human activity habitats and its social systems (Fig. 1, 3) [5]. Applying the concept of the ecosystem, these systems can be study as the software engineering ecosystems, taking into account the additional components of biotic is anthropos and abiotic are antropogenic artifacts that are inherent in the anthroposphere. In the software engineering ecosystem, except for artifacts that consist of assets and tools [13], data, information and knowledge should be considered, as well as the software product as territory for the software ecosystem (Fig. 3).

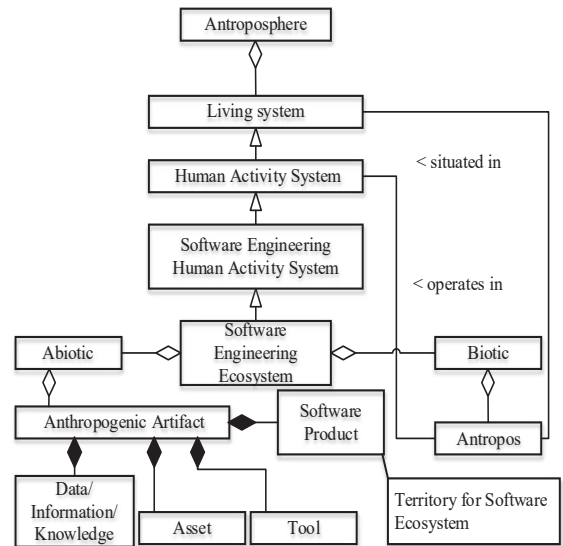


Fig. 3. Software engineering ecosystem

The ontology of the software engineering ecosystem with its attributes is shown on Fig. 4.

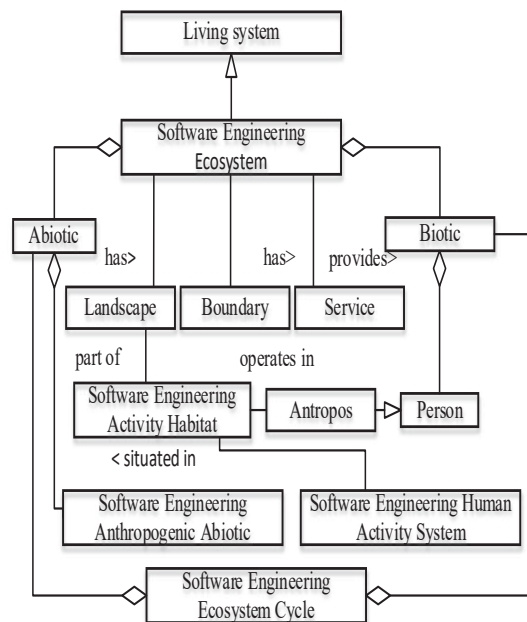


Fig. 4. The ontology of the software engineering ecosystem

In the context of the software engineering habitats the activity of anthropos to lead to implementing processes that can be a linear (chains), nonlinear (networks) and repetitive (cycles). These processes must be objects for the ecosystems study regardless of the software engineering ecosystem type [5]. They are the following chains, cycles and networks (Fig. 5):

- software engineering life cycle, this is the main cycle in software engineering,

therefor inherent in application development human activity.

- cycles of data, information and knowledge, especially inherent in such human activity as software engineering education and research, and software research.

- software reuse cycle, analogous to nutritional cycles, here, these are cycles of reuse of software life cycle artifacts (requirements, design, code, and documentation) and knowledge artifacts.

- value chain/web, this is a value chain and network - an analogue of energy chain in the software engineering ecosystem [9].

- waste cycle is a waste creation and disposal cycle using 3R (Reuse, Rework, and Recovery).

All of the above relates to the application of the ecosystem concept in the context of living systems [14]. In the paper [5], it is

shown that the software ecosystem is the application of the concept of the ecosystem to the software product is the habitat of in the sense of artificial life using analogues of the biotic component of biological ecosystems [4, 15, 16] (Table 1). Here, there are more questions than answers. Therefore, next only software engineering ecosystem research in the context of human activity habitat (living system) will be considered further.

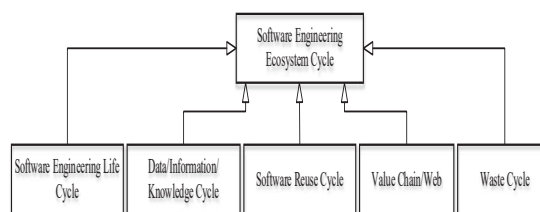


Fig. 5. Software engineering ecosystem chains, cycles and networks

Table 1. Analogues of biotic components of Software ecosystem for biotic components of Biology ecosyste

№	Software ecosystem	Biology ecosystem
1.	Characters, lexemes, simple types	Molecules (?)
2.	Operators, complex types, Structured operators	Tissues
3.	Subroutines (algorithms)	Cells (?)
4.	Modules (Classes), mega modules (Subsystems)	Organs (organisms)
5.	Programs	Genus
6.	Computer programs	Species
7.	Software	Subspecies
8.	Software of domain	Varieties
9.	Software product	Individual of the Varieties
10.	Software products of same type on the domain territory	Population
11.	Software product line (family)	Evolving Population
12.	Populations of the Software products on the some domain territory	Community (Biocenosis)
13.	Populations of the Software products on the some domain territory and abiotic	Biogeocenosis

14.	Software products as systems	Ecosystems
15.	Information flows	Energy flows (?)
16.	Control (algorithms)	DNA, RNK (?)

Ecosystem research

In accordance with the goals of studying objects in software engineering using the concept of the ecosystem, we will use traditional approaches to studying systems [3]. However, need take into account the following features:

- Structural analysis is performed on biotic and abiotic components. In addition, structural analysis will have study the biotic component as a community, including its various forms, such as the working community, the program community, and abiotic component as, for example, the artifacts populations.

- Functional analysis is aimed at studying the added value chains instead of food chains, and circulation chains of artifacts of the legacy software instead of nutritional cycles. For software engineering ecosystems with a human community in the biotic component, and possibly software ecosystems, flows that lead to the emergence and elimination of waste should be investigate.

- System analysis, as in the biological ecosystem, aims to study the emergence, stability, self-organization, evolution and sustainability of the ecosystems.

Ecosystem research in the software engineering

Next, based on research on biological ecosystems, we will consider the main areas of research applying the concept of the ecosystem in the software engineering. The research will be divided into two groups - basic and target (Fig. 6). Basic studies are those aimed at collecting data and information about the ecosystems and are divided on long-term study and the ecosystems history study. Targeted research includes structural, functional and system analysis.

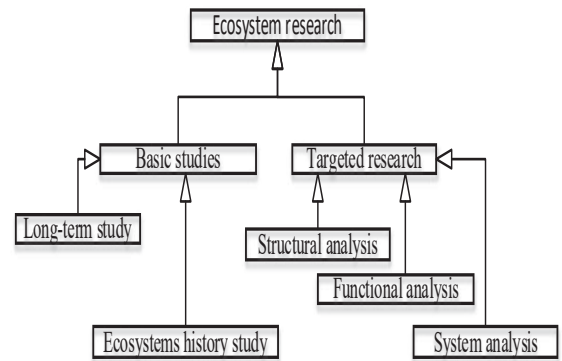


Fig. 6. Software engineering ecosystem research

Basic studies are those aimed at collecting data and information about the ecosystems. This will provide more complete and reliable results of targeted studies. Long-term studies and historical research are the basic research.

Long-term study is research conducted in the ecosystem, continuously for long periods of time [17, 18]. The duration of research should be at least such as long as the dominant event of the ecosystem occurs or until the event under study changes [18]. In article [19], a classification of types of long-term ecological studies is provided. There are retrospective research (passive - deep and recent; active - opportunistic, intentional and remodeled) and prospective research. Research data can be directed to the following: to develop and test the theory; to provide general knowledge about the operation of selected ecosystems; to provide student and postgraduate training [17]. The long-term ecological studies have features that consider before starting research. It is, firstly, permanent providing of money, time, staff, means; secondly, keeping research from complacency, if funding job is provided in the long run; third, limit by practical considerations, for example, by time [17]. Apparently, these features need

take into account for the software engineering ecosystem research. The long-term studies expedient when there are the slow processes, rare events, complex phenomena [17]. The field study and the field experiment can be used for the long-term ecological studies.

The ecosystems history study. As the structure, functions, services, landscape, and terrain of the ecosystems change over time, the ecosystems have a history. For quantitative and especially comparative studies, it is critical to determine the current state of the ecosystem and state changes over time. For example, how the transformation of the ecotope into a biotope of the software engineering ecosystem was carried out. For historical research, the empirical or natural-historical method, the field study, the anthropological and ethnographic studies can be use.

Targeted research includes structural, functional and system analysis.

Structural analysis aims to study the following components of the ecosystem as structure [10]. Firstly, activities, which specify the actions for the value proposition to materialize. Secondly, creating by the actors. Thirdly, positions that determine places where actors are in the flow of activity. Fourthly, links that determine the transfer of value between the actors. The essence of the value may vary - material, information, influence, funds. Need to be considered that there is the transfer of energy in the ecosystem, biotic and abiotic are characterized by diversity and therefore need multidiscipline study, and that the structure of the ecosystem can be multi-level. The field study, the monitoring and classification methods can be used for the structural analysis. The software engineering ecologist must follow an example the ecologist in studying species of both biotic and abiotic component of the software engineering ecosystem [20].

Functional analysis contributes to the identification of functions, services and interactions (streams and chains) aimed at creating value.

Ecosystem functions on the definition of ecosystem processes are based, which can be interactions inside biota and between the biotic and abiotic components of the ecosystem, covering material cycles and energy flows. The functions of the ecosystem can be defined as

the ability of processes and components of the structure to provide goods and services that meet human needs, directly or indirectly. Article [21] examines the historical development of the concept of ecosystem services in economic theory and practice, which can be useful in the software engineering.

Ecosystem services are defined as a set of ecosystem functions that are useful for humans [22]. Thus, ecosystem services in software engineering should include the results of performing ecosystem functions that are beneficial to humans. In the software engineering, as with other types of the ecosystems, these functions can be divided into three groups - economic, sociocultural and environmental [22]. The first group, depending on the type of ecosystem, includes knowledge, skills, products and artifacts. The second group is determined by the presence of people in the biota of the ecosystem that bring socio-cultural values to the concept of the ecosystem. Both are related to not only the culture of software engineering, but also to society and culture in a broad sense. The third group, these are functions regarding the ability of ecosystems to regulate flows that lead to waste disposal [4]. An important aspect in the study of ecosystem services is their evaluation. The papers [23, 24] offer models for evaluating ecosystem services that can be applied to software engineering ecosystems.

Researching flows, chains and cycles that correspond to ecosystem functions is an important part of the functional ecosystem study. For the software engineering ecosystems, these are the study of the added value chains (food chain analogues), reuse cycles of the legacy software artifacts, knowledge, practices (analogues of nutritional cycles). In biological ecosystems, the nutrient cycle is one of the most important processes, as it ensures the transformation of "old" (legacy) matter into nutrients used by living organisms. It also maintains the balance of those nutrients that are necessary to maintain a healthy ecosystem. Biological ecosystems consider four cycles of nutrients (water, oxygen, nitrogen, carbon). In software engineering ecosystems, as an analogue of such cycles, we suggest using the software engineering artifacts (knowledge, policies, software artifacts, best practices) reuse

cycles (Fig. 4). Without reuse cycles, software engineering ecosystems can become unbalanced and unhealthy, similar to biological ecosystems, leading to inefficiency and disintegration. In addition to them, flows that lead to generated and disposal of waste should be studied. Waste cycles are important and arise for various reasons, for example, waste is a result of inappropriate organization of the activity process. For functional analysis, the technique of system functional analysis can be used. For value chains and cycles, it may will be useful to apply value chain analysis [25, 26]. System analysis is aimed at studying a part of the world as a system localized within the ecosystem. Of particular interest are the emergent functions of the system and modeling and simulation. Since ecosystems are usually open systems, connections between the ecosystem and other parts of the world presented as ecosystems are also studied. For system analysis, experimental simulation, simulation experiment, conceptual modeling, simulation should be used. Finally, it should be noted that the ecosystem research differs from systems research in that it is interdisciplinary.

Literature Review

The literature review was performed. The goal of review is find out the state of the ecosystem research in context of the software. For this goal are formulated the following two research questions:

(RQ1) Do software ecosystem researchers use biological analogies, and if so, which ones?

(RQ2) What type of the ecosystem research have software ecosystems researchers shown interest in?

Below are the results of the analysis.

RQ1. Do software ecosystem researchers use biological analogies, and if so, which ones?

In the works [16, 27] are attempts to use analogies. For example, [27] offers two types of biota and an analogue of the food chain for the software ecosystem. The first type is the participants of the software project, which are represented by four categories - core developers, active developers, peripheral developers and users. Between them, a connection is indicated, according to the authors, sim-

ilar to a trophic chain directed from core developers to users. The second type are the software components and the project contributors that are compared whit the biological species in the biological ecosystem. Indicated that as in the trophic network, some of these components (such as joint development platforms and software libraries) act as producents. A wider range of analogues of components for this type can be found in the work [16]. However, it has no analogies with chains and cycles.

RQ2. What type of the ecosystem research have software ecosystems researchers shown interest in?

Basic studies (Long-term studies). The paper [28, 29] is work from this type of research. Author studied three aspects of the software ecosystems - how this ecosystem has emerged; how the present organization works in terms of its structure, its processes, and its product line, and how the organization relates to actors in its external environment.

Basic studies (History of ecosystems). The paper [29] presents the results of studying how the object-oriented classes are used in software packages over time to increase understanding of the evolution of the software ecosystem, as well as the maintenance and survivability of projects.

Targeted research (Structural analysis). The article [30] considers a subset of communities found on GitHub and a variety of roles is identified. The article [31], bots as parts of biotic are used to coordinate open source software projects. Four different classes of the bots that coordinate work in such projects (broker, checker, gatekeeper, and manager) have been identified. In the articles [32, 33], the workforce ecosystem as a structure focused on creating value for the organization was defined.

Targeted research (Functional analysis).

- Ecosystem functions and services. In the article [34], proposed an open source online platform (RECOSystem) to support the development and management of recommender systems. In the article [35], the definition of the software testing ecosystem was proposed. The article [36], described the component-based software development ecosystem and its main service is the supply of compo-

nents to the market. The article [37], described the ecosystem of scientific software, as well as the production of scientific software components. In the article [38], the program analysis platform that aims to build a data-driven ecosystem is presented. The article [39], presents an ecosystem that Ericsson has developed for the systematic practice of large-scale reuse of microservices in a cloud context. The article [40] draws attention to decision-making on the design of software and systems for products, processes, projects that maximize the value created for invested resources. In the article [41], the concept of a unified software value chain and the first empirical proof of the concept are proposed. This article examines the value chain in the context of the software life cycle. This view supports our view of software engineering ecosystems. In the article [42], the concept of the software ecosystem service is discussed.

- The reuse cycles of the legacy software. In the article [43], the concept of the ecosystem applies to the software industry to encourage developers to reuse and multiple use the software components. This view cannot be directly related to the cycles of reuse of the legacy software, but it can be accepted to study these cycles. In the article [44], value networks for DevSecOps are presented.

- Flows of waste generation and disposal. The work [45] is the most famous in the software engineering about this theme. This aspect of the ecosystem research can certainly be related to issues of the sustainable and environmental software engineering [46].

Targeted research (System analysis).

The article [47] the structures of known ecosystems, as well as formulated and analyzes problems related to openness, management, analysis and quality in such ecosystems are presented. The article [48] draws attention to orchestration in software ecosystems and offers a new understanding of it. The article [49] provides a systematic display of the description and documentation of the software ecosystem through modeling.

Conclusion

This work is a continuation of the author's works [4, 5, 50 - 52]. Represents the author's view on the problem of applying the con-

cept of the biology ecosystem in software engineering. Attention is drawn to the inadequacy of the existing application of the concept to the software ecosystem compared to the biological approach. That can be explain by the fact that the software ecosystem uses an approach that considers ecosystems as networks, organized around a keystone species and characterized by a large number of loosely interconnected participants who depend on each other for their mutual effectiveness and survival. In paper [10], such approach was called "ecosystem-as-affiliation", and in paper [11], it was called "platform ecosystems". At the same time, the boundaries of the landscape and itself landscape are not specified and there are no analogues of chains and cycles, the presence of which determines the essence of biological ecosystems. Therefore, now the software ecosystem, in our opinion, is nothing more than a fashionable term [53]. In this article, attention is drawn to a different approach "ecosystem-as-structure", when ecosystem is configuration of activities that are defined by a value proposition. In this context, attention is drawn to the ecosystem research and the need to find analogies that are known in ecology.

References

1. R. V. O'neill, Is it time to bury the ecosystem concept (with full military honors, of course!), *Ecology*, 82(12), (2001) 3275–3284.
2. E. Becker, B. Breckling, Border zone between ecology and systems theory, *Ecology Revisited*, Springer, 2011, 385–403.
3. A. Mittwollen, Unity in Ecology? An Investigation of Patterns, Problems and Unifying Concepts of Population Ecology, Systems Ecology and Evolutionary Ecology (Biology), Ph.D. thesis, Bremen, 2002.
4. N.A. Sydorov, Software ecology, *Software engineering*, 1 (2010), 53-61 (in Ukrainian).
5. N. Sydorov, Software Engineering Ecosystems, in: *Proceedings of the 13th International Scientific and Practical Programming Conference UkrProg 2022Kyiv, Ukraine, October 11-12, 2022*.CEUR-WS: 10-Oct-2023, Vol-3501, pp. 91-100.
6. E P. Odum, *Fundamentals of Ecology*, Saunders Company, Philadelphia-London, 1971.
7. K. Manikas, Software ecosystems – A systematic literature review, *The Journal of Systems and Software* 86 (2013), 1294–1306.

8. O. Barbosa, C. Alves, A Systematic Mapping Study on Software Ecosystems, in: *Proceedings of the Workshop on Software Ecosystems*, 2011, pp. 15-26.
9. R. Costanza, C. J. Cleveland, Value theory and energy, in: *Encyclopedia of energy*, 6, 2004, pp. 337 - 346.
10. R. Adner, Ecosystem as Structure: An Actionable Construct for Strategy, *Journal of Management* Vol. 43 No. 1, January (2017), 39–58.
11. K. Kapoor, A. Ziaee Bigdeli, Y.K. Dwivedi, A socio-technical view of platform ecosystems: Systematic review and research agenda, *Journal of Business Research* 128 (2021) 94–108.
12. D. H. Ackley, Real artificial life: Where we may be, in: *The Proceedings of Artificial Life VII*, Portland, Oregon, August, 2000.
13. E. Zabardast, J. Frattini, Assets in Software Engineering: What are they after all?, *Journal of Systems and Software* 193(2022), 111485.
14. F. Capra, P. L. Luisi, *The systems view of life: a unifying vision* / University Printing House, Cambridge, United Kingdom, 2014.
15. D. H. Ackley, Real artificial life: Where we may be, in: *The Proceedings of Artificial Life VII*, Portland, Oregon, August, 2000.
16. T. N. Nguyen, *The Ecology of Software: A Framework for the Investigation of Business-IT Integration Issues and Trends of Information Technology Management in Contemporary Organizations*, in: *Proceedings of the Information Resources Management Association International Conference*, 2002.
17. D. Strayer, J. S. Glitzenstein, Long-Term Ecological Studies: an Illustrated Account of Their Design, Operation, and Importance to Ecology, in: *Occasional Publication of The Institute of Ecosystem Studies* Number 2, August, 1986.
18. G. E. Likens, *The ecosystem approach: its use and abuse*, Germany, 1992.
19. S. T. Jackson, Repurposing long-term ecological studies for climate change <https://doi.org/10.1073/pnas.2314444120>.
20. D. Kirsh, Explaining Artifact Evolution. in: *Cognitive Life of Things: Recasting the Boundaries of the Mind*. Malafouris, L. (Ed.), McDonald Institute for Archaeological Research, Cambridge, April 7-9, 2009. Cambridge University press.
21. E. Gómez-Baggethun, The history of ecosystem services in economic theory and practice: From early notions to markets and payment schemes, *Ecological Economics*, vol. 69 (2010), 1209-1218.
22. R. S. de Groot, A Typology for the Classification Description and Valuation of Ecosystem Functions, Goods and Services, *Ecological Economics*, *Ecological Economics* Volume 41, Issue 3, (2002) 393-408.
23. L. Hein, K. van Koppen, Spatial Scales, Stakeholders and the Valuation of Ecosystem Services, *Ecological Economics*, 57, (2006), 209–228.
24. J. J. Kaufman, *Stimulating innovation in products and services: with function analysis and mapping*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2006.
25. J. R. Wixson, Function Analysis and Decomposition Using Function Analysis Systems Technique, in: *International Council on Systems Engineering Annual Conference (INCOSE '99)*, 1999.
26. B. Boehm, *A Value-Chain Analysis of Software Productivity Components*, TRW Inc., 1976.
27. T. Mens, P. Grosjean, The Ecology of Software Ecosystems, *Computer*, October (2015), 112-114.
28. G. K. Hanssen, A longitudinal case study of an emerging software ecosystem: Implications for practice and theory, *The Journal of Systems and Software* 85 (2012) 1455–1466.
29. T. Mens, M. Claes, P. Grosjean, *ECOS: Ecological Studies of Open Source Software Ecosystems* (2014), Antwerp, Belgium 978-1-4799-3752-3/14, 2014.
30. D. Mendez, B. Baudry, M. Monperrus, Empirical Evidence of Large-Scale Diversity in API Usage of Object-Oriented Software, in: *Proceedings of IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, July 2013.
31. F. Hukal, N. Berente, Bots Coordinating Work in Open Source Software Projects, *Computer*, 52(9) (2019), 52-60.
32. E.J. Altman, J. Schwartz, D. Kiron, R. Jones, and D. Kearns-Manolatos, *Workforce Ecosystems: A New Strategic Approach to the Future of Work*, MIT Sloan Management Review and Deloitte, April 2021.
33. A. T. Gorski, I. Gligorea, Workforce and workplace ecosystem – challenges and opportunities in the age of digital transformation in: *Proceedings of the International Conference knowledge-based organization*, Vol. XXVIII ,No 1, 2022.
34. A. Abdalla, A Software Ecosystem platform for the development of Recommender Systems, June, 2020, DOI: 10.21203/rs.3.rs-34335/v1.

35. I. Santos, Software testing ecosystems insights and research opportunities, in: Proceedings of the XXXIV Brazilian Symposium on Software Engineering, 2020, pp. 421–426.
36. J. Arndt, J. Dibbern, The Tension between Integration and Fragmentation in a Component Based Software Development Ecosystem, in: Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.
37. J. Howison, Understanding the scientific software ecosystem and its impact: Current and future measures, Research Evaluation 24 (2015), 454–470.
38. C. Sadowski, Tricorder: Building a Program Analysis Ecosystem, in: Proceedings of the 37th IEEE International Conference on Software Engineering, Florence, Italy, May 2015, pp. 598-608, DOI: 10.1109/ICSE.2015.76.
39. M. Usman, D. Badampudi, An Ecosystem for the Large-Scale Reuse of Microservices in a Cloud-Native Context, IEEE Software, (39)5, Sept.-Oct., 2022.
40. B. Boehm, Software Economics: A Roadmap, March 2000, doi: 10.1145/336512.336584.
41. A. Pussep, The Software Value Chain as an Analytical Framework for the Software Industry and Its Exemplary Application for Vertical Integration Measurement, in: Proceedings of the Seventeenth Americas Conference on Information Systems, Detroit, Michigan August 4 -7, 2011.
42. K. Manikas, K. M. Hansen, Software ecosystems – A systematic literature review, the Journal of Systems and Software, 86, (2013), 1294–1306.
43. J. Arndt, J. Dibbern, The Tension between Integration and Fragmentation in a Component Based Software Development Ecosystem, in: Proceedings of the 39th Hawaii International Conference on System Sciences, 2006.
44. J. Morales, R. Turner, Guide to Implementing DevSecOps for a System of Systems in Highly Regulated Environments, Technical Report CMU/SEI-2020-TR-002, 2020.
45. M. Poppendieck, T. Poppendieck, Implementing Lean Software Development: An Agile Toolkit, Addison-Wesley, 2003, 240p.
46. E. Kern, Green software and green software engineering - definitions, measurements, and quality aspects, in: Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich, Feb, 14-16, 2013.
47. J.V. Joshua, Software Ecosystem: Features, Benefits and Challenges, International Journal of Advanced Computer Science and Applications (IJACSA), 8(4), (2013), 242-247.
48. L. M. Maruping, S. Matook, The evolution of software development orchestration: current state and an agenda for future research, European Journal of Information Systems, 29:5 (2020), 443-445, DOI: 10.1080/0960085X.2020.1831834.
49. O. Pettersson, J. Andersson, A Survey of Modeling Approaches for Software Ecosystems, Springer International Publishing Switzerland, 2016, pp. 79-93.
50. N. Sydorov, Programming Style as an Artefact of a Software Artefacts Ecosystem, in: Proceedings of the Advances in Computer Science for Engineering and Education Applications, Springer, Cham, 2021, 244-255.
51. N.A. Sydorov, N. N. Sydorova, E.N. Sydorov, Description model of programming style ecosystem, Problems in programming, 2-3(2022), 74-81.
52. N.A. Sydorov, Toward a software artifacts ecosystem, Problems in programming, 4 (2020), 11.
53. S. Hyrynsalmi, S. M. Hyrynsalmi, Ecosystem: A Zombie Category? 2019 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC).

Одержано: 25.10.2024

Внутрішня рецензія отримана: 01.11.2024

Зовнішня рецензія отриман: 02.11.2024

Про автора:

Сидоров Микола Олександрович,

д.т.н., професор

ORCID 0000-0003-0800-1668

Місце роботи автора:

НТУ України «Київський політехнічний інститут імені Ігоря Сікорського»,

кафедра інформатики

та програмної інженерії,

Моб. тел.: 067 7980361.

E-mail: nyksydorov@gmail/com

ВИМОГИ ДО ОФОРМЛЕННЯ СТАТЕЙ

Журнал «Проблеми програмування» публікує наукові матеріали, які раніше не публікувалися в інших виданнях.

Мова статті: українська, англійська.* Обсяг статті - від 6 до 16 сторінок формату А4. Обсяг матеріалів, що подаються на конференцію УкрПРОГ (кожний парний рік) для публікації у подвійному номері складає від 6 до 8 сторінок формату А4. Автори з України подають матеріали на конференцію українською мовою.

Документ зберігається у форматі doc або docx. Назва файлу включає транслітерацію прізвища автора (авторів), наприклад, "Petrenko.doc".

Автори можуть користуватися електронною поштою для передачі до редакції тексту статті, ділової переписки та правки при коректурі. Е-mail редакції: alengoro@isofts.kiev.ua. Телефон: +380 (96) 418 3082.

1. Оформлення файлу з текстом статті

При підготовці файлу використовуються: стиль нормальний (звичайний) або normal; шрифт Times New Roman, розмір шрифту 12 пт.; міжрядковий інтервал – 1,0; абзацний відступ – 1,25 см; вирівнювання – по ширині. У тексті не допускається вирівнювання пропусками; розстановка переносів – автоматична. Формат паперу А4, розміри полів документа – 20 мм. Текст статті після зазначення авторів, назви і анотацій (двома мовами) має бути оформлений у 2 колонки, ширина яких – 7,86 см, а пробіл між ними – 1,27 см.

2. Розміщення матеріалу статті

Верхній колонтитул: назва рубрики відповідно до переліку, прийнятому редакцією журналу (*пропонується авторами, остаточно узгоджується редакцією*).

УДК (зліва під рискою верхнього колонтитулу): індекс за універсальною десятковою класифікацією; **DOI:** в тому ж рядку правіше (*заповнюється редакцією*).

Автори: ініціали та прізвища авторів, курсив (світлий).

Заголовок 1 (назва статті): не містить аббревіатур та строго відповідає змісту статті. Шрифт 15 пт, напівжирний, регістр верхній.

Анотація: 50-100 слів, не містить аббревіатур, не зрозумілих зі змісту статті. Шрифт 10 пт, звичайний.

Ключові слова: не більше 10 слів, не містить аббревіатур, не зрозумілих зі змісту статті, подаються в називному відмінку, розділені комами. Шрифт 10 пт, звичайний.

УВАГА! Автори, заголовок статті, анотація і ключові слова зазначаються **ДВІЧІ:** українською і англійською мовами. Спочатку мовою статті, потім іншою мовою.

Нижній колонтитул (тільки для першої сторінки) включає стандартну інформацію сорту right: перший рядок – прізвища авторів, рік; другий рядок – номер ISSN, назва журналу, рік, номер випуску.

Заголовок 2 (назва розділу): шрифт 14 пт, напівжирний; абзац із центральним вирівнюванням, без переносів. Заголовки нижчого рівня (пункти і т.п.) у самостійний абзац не виділяються і проходять першим реченням текстового абзацу, шрифт 12 пт, напівжирний.

Основний текст статті (виступ на конференції) включає такі необхідні елементи:

постановка проблеми в загальному вигляді і її зв'язок з важливими науковими або практичними завданнями;

аналіз останніх досліджень і публікацій, у яких розпочато рішення даної проблеми і на які спираються автори, виділення невирішених раніше частин загальної проблеми, яким присвячується дана стаття;

формулювання цілей статті (постановка задачі виступу на конференції);

виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів;

висновки з даного дослідження і перспективи подальших розробок у даному напрямку; подяка (за наявності такої).

Формули створюються в редакторі Microsoft Equation 3.0 або MathType. Формули, на які є посилання в тексті, повинні мати наскрізну нумерацію. Номер формули друкується в круглих дужках біля краю правого поля. Розмір основного шрифту редактора формул – 12 пт. Розміри символів у формулах: звичайний – 12 пт, великий індекс – 9 пт, дрібний індекс – 7 пт, великий символ – 18 пт, дрібний символ – 11 пт. Не допускається масштабування формульних об'єктів.

Рисунки мають бути створені вбудованим редактором Word Picture або експортовані з прикладних програм Windows у графічних форматах (bmp, psx, gif, jpg або tif). Рисунки розташовуються по центру. Нумерація рисунків здійснюється відповідно до порядку згадування у тексті. Нумеровані підписи розміщуються під рисунком з позначенням «Рис. », далі вказується номер рисунка і текст підпису.

Таблиці мають бути підготовлені стандартним вбудованим в Word інструментарієм «Таблиця». Таблиці нумеруються за порядком згадування. На номер таблиці повинно бути посилання в тексті. Номер таблиці вказується в окремому рядку з вирівнюванням по правій стороні (наприклад, «Таблиця 1»). Назви таблиць розміщуються над таблицею з вирівнюванням по центру. Мінімальний розмір шрифту в таблицях – 11 пт.

Література: нумерований список джерел згідно ДСТУ 8302:2015 від 01.07.2016 р., шрифт 11 пт, відступ: спеціальний, навислий, 0,63 см. Джерела з заголовками на латиниці наводяться без перекладу. Інші джерела подаються мовою оригіналу. Приклади оформлення бібліографічних посилань згідно з вимогами **Harvard Style** наведені в багатьох публікаціях, наприклад: http://www.staffs.ac.uk/assets/harvard_referencing_examples_tcm44-39847.pdf

Дата надходження статті (матеріалів конференції) позначається цифрами окремим рядком після слова «Одержано:»/”Received:”.

Відомості про рецензентів зазначаються редакцією після слова «Рецензенти:».

Дані про авторів: мають починатися рядком “Про авторів:”, напівжирний курсив. Далі вказуються для кожного з авторів ПІБ повністю, вчений ступінь, наукове звання, посада, обов’язково номер ORCID (сайт ORCID <http://orcid.org/>).

Дані про місце роботи авторів: починаються рядком “Місце роботи авторів:”, напівжирний курсив. Далі вказуються місце роботи, адреса, телефон, електронна пошта, контактний телефон особи, відповідальної за зв’язок з редактором.

Для уникнення неузгодженості перелік авторів подається під номерами, що відповідають нумерації закладів, де вони працюють.

3. Оформлення файлу з анотаціями

Файл з анотаціями містить інформацію двома мовами – англійською і українською та має бути оформлений у дві колонки: УДК (шрифт – 8 пт); нижче - назва статті (шрифт – 12 пт, напівжирний); прізвища та ініціали авторів (шрифт – 12 пт); текст анотації, ключові слова (шрифт – 10 пт).

Вимоги до анотації: обсяг від 100 до 250 слів, інформативність, оригінальність, змістовність (відображає основний зміст статті і результати досліджень), структурованість (дотримується логіки опису результатів у статті).

Документ зберігається у форматі doc або docx. Назва файлу подається транслітерацією прізвища автора (авторів), наприклад, “Petrenko_Annot.doc”.

*16.07.2020 р. набули чинності положення Закону України «Про забезпечення функціонування української мови як державної». Відповідно до статті 22 «Державна мова у сфері науки» у наукових виданнях не повинно бути вміщено матеріалів іншими мовами, окрім державної, англійської та мов ЄС.

Примітка: Підписний індекс журналу «Проблеми програмування» – **90853**.