

О.В. Палагін, М.Г. Петренко, А.А. Літвін, М.О. Бойко

ПРО ОДИН ПІДХІД ДО АВТОМАТИЧНОГО СТВОРЕННЯ ФОРМАЛЬНИХ ЗАПИТІВ ДО ОНТОЛОГІЧНИХ БАЗ ЗНАНЬ

Пропонований у статті підхід передбачає аналіз коротких природномовних повідомлень, поданих українською мовою, та автоматичне формування на їхній основі запитів мовами SPARQL і Cypher. Як засіб оброблення SPARQL-запитів слугує сервер Apache Jena Fuseki, а сховища даних або онтологічної бази знань – графова база даних Neo4J. Остання є найпоширенішою базою даних із відкритим вихідним кодом, високопродуктивною і добре масштабованою, тобто здатною працювати з великими обсягами даних. Крім того підходи до побудови формальних запитів на базі природномовних запитів для мови Cypher маловідомі і потребують подальшої розробки. Підхід базується на тому, що природномовний запит користувача піддається низці послідовних перевірок. Їх результати визначають набір семантичних типів, виражених у фразі (природномовного запиту), та відповідні поняття, що їх визначають. Результатом вказаних перевірок є набір чотирьох значень – кодів результатів перевірок, а також підметів і присудків, якщо вони присутні. Цієї інформації достатньо для вибору набору базових шаблонів для формальних запитів. За результатами таких базових перевірок створюються основні базові шаблони для формування прикінцевого запиту. У запропонованому підході є базовий шаблон запиту, спрямований на отримання інформації певного типу у заданій формі, а також додаткові шаблони-модифікатори, які опціонально конструюють рядки запиту у відповідних блоках основного запиту, вводячи додаткові умови. Описано процес автоматичної генерації SPARQL-запитів до контекстної онтології на прикладі бази знань медичних статей з рецензованих журналів з відкритим доступом. Особливість підходу полягає у тому, що формальний запит автоматично будується із блоків шаблонів (основних і допоміжних), які налаштовуються, відповідно до визначених семантичних категорій, присутніх в аналізованому тексті, та сутностей, що їх конкретизують.

Ключові слова: технологія Semantic Web, онтологічна база знань, OWL онтологія, мови SPARQL і Cypher, графова база даних Neo4J.

О.В. Palagin, M.G. Petrenko, A.A. Litvin, M.O. Boyko

ABOUT ONE APPROACH TO AUTOMATIC CREATION OF FORMAL QUERIES TO ONTOLOGICAL KNOWLEDGE BASES

The article develops an approach that includes the analysis of short natural language messages in Ukrainian and the automatic generation of queries in SPARQL and Cypher based on them. The Apache Jena Fuseki server is used as a SPARQL query processing tool, and the Neo4J graph database is used as a data warehouse or ontological knowledge base. The latter is the most common open source database, high-performance and well-scalable, i.e., capable of working with large amounts of data. In addition, approaches to building formal queries based on natural language queries for Cypher are little known and require further development. The approach is based on the fact that a user's natural language query is subjected to a series of sequential checks. Their results determine the set of semantic types expressed in the phrase (natural language query) and the corresponding concepts that define them. The result of these checks is a set of four values – the codes of the check results, as well as the subjects and predicates, if present. This information is enough to select a set of basic templates for formal queries. Based on the results of such basic checks, the main basic templates for generating the final request are created. The proposed approach has a basic query template aimed at obtaining information of a certain type in a given form, as well as additional modifier templates that optionally construct query strings in the corresponding blocks of the main query by introducing additional conditions. The article describes the process of automatic generation of SPARQL queries to a contextual ontology using the example of a knowledge base of medical articles from peer-reviewed open access journals. The peculiarity of the approach is that the formal query is automatically built from blocks of templates (main and auxiliary), which are customizable in accordance with certain semantic categories present in the analyzed text and the entities that specify them.

Keywords: Semantic Web technology, ontological knowledge base, OWL ontology, SPARQL and Cypher languages, Neo4J graph database.

Вступ

Розроблення застосунків, заснованих на технологіях Semantic Web, Big Data, Natural Language Processing у поєднанні з неймережевими технологіями де-факто стало одним із найактуальніших напрямків наукових досліджень і практичних розробок. Зокрема, це стосується і побудови онтологічних систем і відповідних баз знань, цікавих для користувачів.

Опрацювання інформації та представлення знань на основі онтологій виникло в результаті пошуку стандартного протоколу для впорядкування знань у різноманітних сферах знань. Ця парадигма має на меті запропонувати уніфіковану схему та основні принципи для систематичного представлення, категоризації та взаємозв'язку знань, незалежно від галузі знань. Поява онтологічних стратегій уможливила ефективну побудову знанняорієнтованих систем і, що особливо важливо, заклала фундамент для трансдисциплінарної взаємодії та онтологічного інжинірингу у сфері сучасного штучного інтелекту [1–3].

Характерною рисою людського інтелекту є здатність асимілювати інформацію з одного джерела і адаптувати її в різних сферах, що лежить в основі творчості та інновацій. Для того, щоб універсальний машинний інтелект був практично ефективним, він повинен виходити за межі простого розуміння тексту. Його справжня перевага полягає в здатності використовувати свій запас знань для вирішення нових завдань. Вміння системи штучного інтелекту застосовувати знання в різноманітних і нових сценаріях цілком може стати визначальним критерієм для оцінки його інтелектуальної глибини [4].

Нами були виконані розробки згаданої вище онтологічної системи, описаної в [5–7]. Вона значно пришвидшує отримання наукової інформації користувачу, але її слабкістю було ручне або автоматизоване створення бази даних наукових публікацій і SPARQL запитів. Спілкування користувача з базою знань природною мовою передбачає використання формальних мов запитів. Отже, під час створення діалогових систем з природно-

мовним інтерфейсом виникає потреба в автоматичній генерації пакетів формальних запитів на основі природномовних запитів користувачів. У статті розглянуто створення запитів мовою SPARQL та все більш перспективною мовою запитів Cypher, яка використовується в графовій базі даних Neo4J. Важливо зазначити, що наразі створення запитів мовою Cypher на основі природномовних фраз є недостатньо дослідженим, в тому числі для української мови. Тому цей напрямок досліджень і досі актуальний, особливо для української мови.

Створення формальних запитів на основі аналізу природномовних повідомлень користувачів

Запропонований підхід базується на тому, що природномовний запит користувача піддається низці послідовних перевірок. Їх результати визначають набір семантичних типів, виражених у фразі (природномовного запиту), та відповідні поняття, що їх визначають. Схема спрощеної й ефективної версії цього підходу наведена на рис. 1.

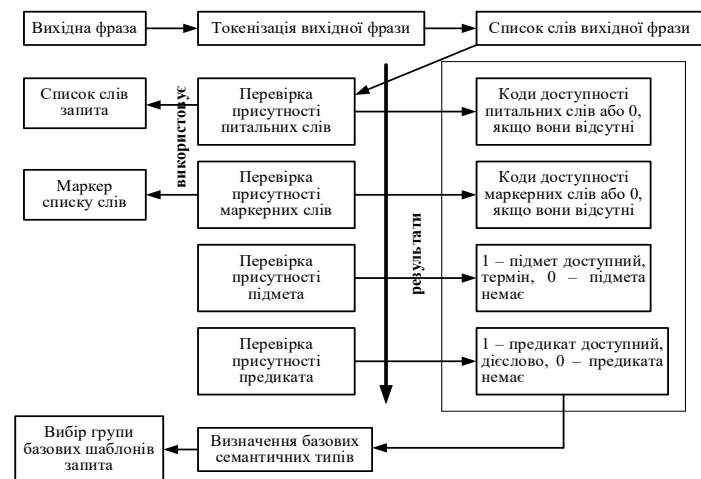


Рис. 1. Схема розбору фрази користувача для вибору базового набору формальних шаблонів запитів

Відзначимо особливості флективних мов, зокрема, української мови. В ній порядок слів у реченні є менш важливим,

ніж для нефлективних мов, а наявність конкретних слів та їхніх словоформ є важливішою. На Рис. 1 показано схему визначення набору базових семантичних типів і відповідних первинних шаблонів запитів. Для цього оброблений природномовний вираз, токенізований до рівня списків речень і відповідних слів, проходить чотири послідовні перевірки:

Перевірка наявності питальних слів (критичний момент для визначення типу інформації, що запитується).

Перевірка на наявність слів-маркерів, насамперед дієслів, таких як “перебуває”, “працює”, “стоїть”, “надсилає” тощо. Ці слова об’єднуються в групи синонімів зі схожим семантичним забарвленням.

Перевірка наявності підмета, якщо він присутній.

Наступним важливим аспектом після підмета є *перевірка наявності осмисленого присудка*, якщо він не підпадає під категорії слів з *перевірки на наявність слів-маркерів*.

Далі необхідно виконати додаткові перевірки для визначення більш значущих семантичних відтінків. Однак для цих перевірок не існує окремих базових шаблонів, інакше кількість шаблонів значно зростає, а власне шаблони мали б значне дублювання коду. Натомість за результатами цих додаткових перевірок до базових шаблонів вносяться модифікації (зміни та доповнення) згідно з відповідними інструкціями. Це дозволяє зробити підхід більш гнучким і таким, що спрощує як сам процес автоматичного аналізу, так і побудову відповідного програмного комплексу.

Коротко опишемо графову систему керування базою даних (СКБД) Neo4J і мову запитів Cypher до неї. Окрім СКБД, що працюють у зв’язці з OWL/SPARQL механізмами, такими як Jena Fuseki, що де-факто наразі є стандартом, існують і альтернативні підходи до графових баз даних, які також можуть бути використані для зберігання й опрацювання з онтологією. СКБД Neo4J [8] забезпечує досить високу продуктивність і масштабованість, а також придатний для роботи з великими обсягами даних. Мовою формальних запитів,

прийнятою в Neo4J, є Cypher. Вона досить потужна, гнучка та відкрита для розширення функціональності через плагіни, зокрема, для реалізації типових алгоритмів на графах. Однак на даний момент, на відміну від SPARQL [9–11], існує не так багато розробок для перетворення запитів природною мовою у формальні запити на Cypher. Розглянемо детальніше запити, що описуються цією мовою.

XML-шаблони для запитів мовою Cypher

Шаблони запитів мовою Cypher зберігаються у вигляді XML-файлу з певною структурою. Нижче наведено приклад основного шаблону для мови запитів Cypher. Її застосування переважає через значний розмір онтології [12–14], створеної шляхом повністю автоматичного синтаксично-семантичного аналізу тексту [15, 16].

Формування онтології відбувалося згідно методу, описаному в [8]. На основі визначених під час аналізу тексту синтаксично-семантичних зв’язків між поняттями у реченнях будується онтологічна графова структура. Контексти речень і їхні частини також зберігаються у створюваній OWL-онтології. Із цими реченнями поєднані набори наявних у них семантичних зв’язків, конкретизованих відповідними сутностями. Типізація семантичних категорій укладена у встановлену ієрархічну структуру, опис якої подається у [12]. В ній приклад онтології використано як опис підходу побудови запитів мовою Cypher у певній предметній галузі.

Як робочий приклад OWL-онтології контекстів, створеної на основі набору документів із наперед визначеною структурою, використовувалася онтологія з медичної реабілітації на основі файлів наукових статей.

Розглянемо приклад одного з таких шаблонів:

```
<template>
<verbose_name>Common information</verbose_name>
<id>1</id>
<type>base</type>
<variables>
    <variable>
```

```

<name>INPUT_VALUE_1</name>
  <destination>input</destination>
</variable>
<variable>
  <name>CONTEXT</name>
  <destination>output</destination>
</variable>
</variables>
<match>
  (inp:Class)-[]-(n:Relationship),
  (n:Relationship)-[]-(x:Class),
  (n)-[:SPO]->(rel_group),
  (rel_group)-[:SPO]->(rel_sent),
  (rel_sent)-[:SPO]-(sent_super)
</match>
<where>
  inp.label = "INPUT_VALUE" and
  sent_super.name = "SentenceGroups"
</where>
<return>
  DISTINCT rel_sent.label as CON-
TEXT;
</return>
</template>

```

Секції XML-шаблону, а саме `<match>`, `<where>` і `<return>`, відповідають певним секціям формального запиту мовою Cypher [8]. Певні фрагменти вмісту (тексту) в цих секціях слугують шаблонами змінних. Змінні описуються в розділі `<variables>`, де кожна змінна визначається своїм ім'ям – `<name>` і місцем призначення – `<destination>`. Призначення може мати значення або “вхідні” – вказуючи на значення, що підставляються в шаблон, або “вихідні” – вказуючи на змінні, що не замінюються при формуванні запита на конкретні вихідні значення. Натомість вони слугують посиланнями на назви та кількість параметрів, значення яких отримуються під час виконання запиту. Тег `<id>` для ідентифікатора шаблону слугує для зіставлення його з результатом аналізу фрази користувача, а також з відповідним шаблоном для формування відповіді. Тег `<verbose_name>` включено винятково для розпізнавання шаблонів запитів розробником під час розробки та супроводу системи.

Процес автоматичного формування запитів за шаблонами

Розглянемо структуру формальних запитів та спосіб їх формування. Структура онтології дозволяє здійснювати цілеспрямований пошук як контекстів, так і ок-

ремих понять, враховуючи наявність цих понять у контексті та їхню пов'язаність за певним критерієм семантичного типу. У запропонованій схемі є базовий шаблон запиту, спрямований на отримання інформації певного типу в заданій формі, а також додаткові шаблони-модифікатори, які опціонально конструюють рядки запиту у відповідних блоках основного запиту, вводячи додаткові умови.

У мові Cypher запити поділяються на три основні блоки: MATCH, WHERE і RETURN. Блок MATCH задає схему зв'язків між вершинами орієнтованого графа. Блок WHERE накладає умови на властивості (характеристики) вершин і/або зв'язків, зазначених у блоці MATCH. Блок RETURN вказує, що повинно бути виведено в результаті і під яким ім'ям (псевдонімом). В даному випадку це конкретний клас, позначений змінною “inp”. У блоці WHERE нами закладено умову, за якою властивість label вузла “inp” має дорівнювати поняттю, що запитується, (тут і далі в шаблонах запитів INPUT_VALUE представляє текст вхідного поняття). У блоці MATCH вказується, що “inp” – це вузол (укладений у круглі дужки) типу Class. Вона пов'язана з іншою вершиною ‘n’, яка має тип Relationship (властивість в OWL). Тип зв'язку не визначено (квадратні дужки порожні), а напрямок зв'язку не вказано. Це означає, що він може бути прив'язаний як до ДОМЕНУ, так і до ДІАПАЗОНУ. Вказувати напрямок не потрібно, оскільки відомо, що такі зв'язки створюються від властивості до класу. Додатково вказано, що ця властивість також повинна бути пов'язана з певним класом “x”. Потім вказано, що властивість, яка об'єднує ці класи, має бути пов'язана з певним реченням “rel_sent”. Умова “sent_super.name = “SentenceGroups”” гарантує, що “rel_sent” дійсно є реченням. У результаті запитується наступне: вивести “rel_sent.label”, який містить контекст речення під псевдонімом CONTEXT.

Детальніше модифікатори шаблонів – фрагменти, які додаються до основних шаблонів запитів, розглядаються в [12–14]. Там же розглядається процес автоматичної генерації SPARQL запитів до

контекстної онтології на прикладі бази знань медичних статей із рецензованих журналів з відкритим доступом.

Коротко опишемо *реалізацію створення OWL-онтології* [13, 17, 18]. Для реалізації створення бази знань у формі OWL онтології у форматі RDF/XML були розроблені спеціальні скрипти мовою Python. Процес складається з двох етапів.

1. Автоматизоване створення JSON представлення вхідних файлів статей.

2. Формування OWL-онтології. На цьому етапі з використанням отриманого набору структур JSON формується OWL-онтологія. Ієрархічна структура ключів словника JSON формує основу майбутньої системи OWL класів, тоді як відповідні контекстні значення стають іменованими сутностями у своїх відповідних класах. Кожне ім'я файлу статті перетворюється на іменовану сутність у класі «Articles». OWL властивість «Зв'язати зі статтею» встановлює зв'язки між контекстами та відповідними статтями, в яких вони з'являються. Іменовані сутності, визначені в контекстах, також перетворюються на іменовані сутності в класі «Word» і пов'язуються з відповідними контекстами за допомогою OWL властивості «Зв'язати з контекстом». Ця структура дозволяє вибирати певні контексти в онтології за допомогою SPARQL запитів.

Як вказувалося вище, запити користувача до великої за розмірами онтології виконуються надто повільно. Щоб прищвидшити процес отримання відповіді нами розглядається можливість використання апаратних засобів на базі програмовних логікових інтегральних схем [19–21].

На завершення представимо напрями розвитку знання-орієнтованих систем та їх застосунків, які сьогодні є актуальними і перспективними з точки зору загального розгляду наукових знань та їх ефективного практичного застосування – у створенні інноваційних технологій.

По-перше, аналіз онтологічних контекстів у будь-якій предметній галузі дає можливість вибудовувати часову траєкторію процесу формування вторинних знань на основі первинних, і таким чином розробити ефективну технологію побудови

нових знань і на їхній базі нових інноваційних технологій.

По-друге, виходячи з парадигми трансдисциплінарного розвитку науки, перспективним вбачається використання викладених у роботі функціональностей для формування перспективних кластерів конвергенції наукових дисциплін та відповідних технологій [22].

Найближчою задачею може стати формування ефективного інструментарію наукового дослідника, в тому числі для орієнтації на теренах власних публікацій у предметній галузі і порівняння з існуючими в інформаційному просторі [6–7].

Безумовно, побудова інтелектуалізованих довідкових систем у предметних галузях (включно із згаданою медико-реабілітаційною) є прямим продовженням виконаних авторами досліджень. Одним з актуальних застосувань таких систем є створення комфортних умов для управління базами знань з боку широкого кола непрофесійних (з точки зору інформаційних технологій) користувачів.

Насамкінець, не можна не згадати задачу формування мовно-онтологічної картини світу в рамках загальної еволюційної програми і формування планетарної свідомості сучасного покоління [22].

Крім того, важливо зазначити, що в [23] мова йде про проблему публікацій здобутків українських учених в умовах воєнного стану збройної агресії РФ в рейтингових журналах.

Висновки

У роботі розглянуто підхід до побудови формальних запитів до онтології, сформованої автоматично на основі природномовного тексту українською мовою. На основі визначених під час аналізу тексту синтактико-семантичних зв'язків між поняттями у реченнях будується онтологічна графова структура. Контексти речень і їхні частини також зберігаються у створюваній OWL онтології. Із цими реченнями поєднані набори наявних у них семантичних зв'язків, конкретизованих відповідними сутностями. Типізація семантичних категорій укладена у встановлену ієрархічну

структуру. Вказаний приклад онтології використано в процесі опису підходу побудови запитів мовою Cypher. Особливість підходу полягає в тому, що формальний запит автоматично будується із блоків шаблонів (основних і допоміжних), які налаштовуються відповідно до визначених семантичних категорій, присутніх в аналізованому тексті, та сутностей, що їх конкретизують.

Подяка

Дослідження виконано за підтримки гранту Національного Фонду Досліджень України за проектом 2021.01/0136 (2022–2024, проєкт у процесі реалізації) “Розробка хмарної платформи пацієнт-центричної телереабілітації онкологічних хворих на основі математичного моделювання” [24–28], на базі Інституту кібернетики ім. В.М. Глушкова Національної академії наук України, м. Київ, Україна.

Література

- Gomez-Perez A., Fernandez-Lopez M., Corcho O. *Ontological Engineering. Advanced Information and Knowledge Processing*. Springer-Verlag, London, 1 edition, 2004. ISBN 978-1-85233-551-9. DOI: [10.1007/b97353](https://doi.org/10.1007/b97353).
- Studer R. Staab S., editor. *Handbook on Ontologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2 editions, 2009. ISBN 978-3-540-70999-2. DOI: [10.1007/978-3-540-92673-3](https://doi.org/10.1007/978-3-540-92673-3).
- OntoChatGPT information system: Ontology-driven structured prompts for ChatGPT meta-learning / Palagin O., Kaverinskiy V., Litvin A., Malakhov K. *International Journal of Computing*, 22(2):170–183, July 2023. ISSN 2312-5381, 1727-6209. DOI: [10.47839/ijc.22.2.3086](https://doi.org/10.47839/ijc.22.2.3086).
- Ford M. *Rule of the Robots: How Artificial Intelligence Will Transform Everything*. Basic Books, New York, first edition, 2021. ISBN 978-1-5416-7473-8.
- Malakhov, K., Petrenko, M., Cohn, E. (2023). Developing an ontology-based system for semantic processing of scientific digital libraries. *South African Computer Journal*, 2023 35(1), 19–36. <https://doi.org/10.18489/sacj/v35i1.1219>.
- Proceedings of the 13th International Scientific and Practical Programming Conference UkrPROG 2022. Kyiv, Ukraine, October 11–12, 2022. URL: <https://ceur-ws.org/Vol-3501/s26.pdf>.
- M.G. Petrenko, O.V. Palagin, M.O. Boyko, S.M. Matveyshyn. Knowledge-Oriented Tool Complex for Developing Databases of Scientific Publications and Taking into account Semantic Web Technology. *Control Systems and Computers*, 2022, Issue 3 (299), pp. 11–28. DOI: <https://doi.org/10.15407/csc.2022.03.011>.
- Goel A. *Neo4J Cookbook*. Birmingham: Pact Publishing Ltd. May 28, 2015., 1st Edition. 226 P. ISBN-13: 9781783287253. DOI: <https://www.packtpub.com/en-sk/product/neo4j-cookbook-9781783287253?type=print>
- Bob DuCharme. *Learning SPARQL. Querying and Updating with SPARQL 1.1 (Second edition)*, O’Reilly Media, All rights reserved, August 2013. 367p.
- Khan A. Knowledge graphs querying. *ACM SIGMOD Record*. 2023. Vol. 52, N 2. P. 18–29. <https://doi.org/10.1145/3615952.3615956>.
- A Dictionary of Computer Science. Oxford University Press. Seventh edition. Editors for this edition Andrew Butterfield BA, BAI, PhD, Gerard Ekembe Ngondi BEng, MSc. 1211 P. <https://doi.org/10.1093/acref/9780199688975.001.0001>.
- A. Litvin, V. Velychko, and V. Kaverinsky. A new approach to automatic ontology creation from the untagged text on the natural language of inflective type, *Proceedings of the International conference on software engineering “Soft Engine 2022”*, NAU, Kyiv Ukraine, 2022, pp. 37 – 45.
- A. Litvin, V. Velychko, and V. Kaverinsky. A New Approach to Automatic Ontology Generation from the Natural Language Texts with Complex Inflection Structures in the Dialogue Systems Development, *CEUR Workshop Proceedings, 2023, Vol. 3501. pp. 172–185*. <https://ceur-ws.org/Vol-3501/s16.pdf>.
- Kaverinsky, V., Malakhov, K. Natural Language-Driven Dialogue Systems for Support in Physical Medicine and Rehabilitation, *South African Computer Journal*, 2023, Vol. 35, No. 2, pp. 119 – 126. DOI: <https://doi.org/10.18489/sacj.v35i2.17444>.
- A. Litvin, V. Velychko, and V. Kaverinsky. Method of information obtaining from ontology on the basis of a natural language phrase analysis, in: *CEUR Workshop Proceedings, CEUR-WS, Kyiv, Ukraine, 2020: pp. 323–*

330. URL: https://ceur-ws.org/Vol-2866/ceur_322_330_litvin_velichko.pdf.
16. O.V. Palagin, M.G. Petrenko, S. Yu. Svitla, V.YU. Velychko. About one approach to analyzing and understanding natural language objects. *Computer tools, networks and systems*. 2008, №7. pp.128–137.
 17. O. Palagin, V. Kaverinsky, A. Litvin, and K. Malakhov. Ontology-driven development of dialogue systems, *South African Computer Journal*. – Vol. 35. No. 1. – 2023. – P. 37 – 62. DOI: <http://dx.doi.org/10.18489/sacj.v35i1.1233>.
 18. Petrenko, N.G. Computer ontologies and ontology-driven architecture of information systems. Book “Information Models of Knowledge”, ITHEA, Kiev, Ukraine – Sofia, Bulgaria, 2010, pp. 86–92.
 19. Kurgaev, A. F., & Petrenko, M. G. (1995). Processor structure design. *Cybernetics and Systems Analysis*, 31(4), 618–625. DOI: <https://doi.org/10.1007/BF02366417>.
 20. Petrenko, M., & Sofiyuk, A. (2003). On one approach to the transfer of an information structures interpreter to PLD-implementation. *Upravlyayushchie Sistemy i Mashiny*, 188(6), pp. 48–57. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0442276898&partnerID=40&md5=44974b40409363e5fe4378e240149c52>
 21. Petrenko, M., & Kurgaev, A. (2003). Distinguishing features of design of a modern circuitry type processor. *Upravlyayushchie Sistemy i Mashiny*, 187(5), 16–19. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0347622333&partnerID=40&md5=7283307afdf891445ec9062c7b2ff80a>
 22. Alexander V. Palagin, Mykola N. Petrenko. Methodological Foundations for Development, Formation and IT-support of Transdisciplinary Research // *Journal of Automation and Information Sciences*, Volume 50, 2018, Issue 10, PP. 1–17, DOI:<https://doi.org/10.1615/JAutomatInfSci.n.v50.i10.10>.
 23. Inefuku, H., Malakhov, K., Cohn, E. R., & Collister, L. B. (2023). Service Diversification, Connections, and Flexibility in Library Publishing: Rapid Publication of Research from Ukraine in Wartime. *Case Studies in Library Publishing*, 1(1). <https://cslp.pubpub.org/pub/084se42n/release/1>
 24. Malakhov, K. S. (2023a). Insight into the Digital Health System of Ukraine (eHealth): Trends, Definitions, Standards, and Legislative Revisions. *International Journal of Telerehabilitation*, 15(2), 1–21. DOI: <https://doi.org/10.5195/ijt.2023.6599>
 25. Malakhov, K. S. (2023b). Letter to the Editor – Update from Ukraine: Development of the Cloud-based Platform for Patient-centered Telerehabilitation of Oncology Patients with Mathematical-related Modeling. *International Journal of Telerehabilitation*, 15(1), 1–3. DOI: <https://doi.org/10.5195/ijt.2023.6562>
 26. Palagin, O. V., Malakhov, K. S., Velychko, V. Yu., & Semykopna, T. V. (2022). Hybrid e-rehabilitation services: SMART-system for remote support of rehabilitation activities and services. *International Journal of Telerehabilitation, Special Issue: Research Status Report – Ukraine*, 1–16. <https://doi.org/10.5195/ijt.2022.6480>
 27. Malakhov, K. (2022). Letter to the Editor – Update from Ukraine: Rehabilitation and Research. *International Journal of Telerehabilitation*, 14(2), 1–2. <https://doi.org/10.5195/ijt.2022.6535>
 28. O. Palagin, V. Kaverinsky, M. Petrenko, and K. Malakhov, Digital Health Systems: Ontology-based Universal Dialog Service for Hybrid E-rehabilitation Activities Support, The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Dortmund, Germany, 2023. – P. 84 – 89. DOI: <http://dx.doi.org/10.1109/IDAACS58523.2023.10348639>

Одержано: 10.04.2024

Внутрішня рецензія отримана: 17.04.2024

Зовнішня рецензія отримана: 24.04.2024

Про авторів:

Палагін Олександр Васильович,
доктор технічних наук,
академік НАНУ,
заступник директора ІК НАН України.
<https://orcid.org/0000-0003-3223-1391>.

Петренко Микола Григорович,
доктор технічних наук,
провідний науковий співробітник.
<https://orcid.org/0000-0001-6440-0706>.

Літвін Анна Андріївна

молодший науковий співробітник.

<https://orcid.org/0000-0002-5648-9074>

Бойко Микола Олександрович,

науковий співробітник.

<http://orcid.org/0000-0003-1723-5765>.

Місце роботи авторів:

Інститут кібернетики НАН України,

E-mail: incyb@incyb.kiev.ua,

www.incyb.kiev.ua