

А. Долгий, І. Шубін

ДОСЛІДЖЕННЯ СУЧАСНИХ МЕТОДІВ ТА ІНСТРУМЕНТІВ ОНТОЛОГІЧНОГО ІНЖИНІРИНГУ В КОНТЕКСТІ СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ

Предметом статті є сучасні методи та інструменти онтологічного інжинірингу в контексті розроблення інтелектуальних систем, зокрема онтологічний інжиніринг як процес створення формалізованих баз знань для інтелектуальних систем. У статті аналізуються мови онтологічного моделювання (*RDF*, *OWL*, *SWRL* тощо), мова запитів *SPARQL* та інструментарії, такі як *Protégé*, *Hozo* та інші, що дають змогу реалізовувати зазначений підхід. **Мета дослідження** полягає у визначенні та аналізі сучасних інструментів і методів онтологічного інжинірингу для розроблення баз знань, що забезпечують високу точність прогнозування й адаптивність у складних інтелектуальних системах із використанням штучного інтелекту чи іншого алгоритму передбачення. **Завданням цього дослідження** є комплексний аналіз сучасних методів та інструментів онтологічного інжинірингу, зокрема порівняльна характеристика їх можливостей та обмежень у застосуванні в інтелектуальних системах. Дослідження передбачає вивчення методів автоматизації процесів створення та поповнення онтологій, зокрема з використанням глибокого навчання та оброблення природної мови, а також оцінювання перспективних мов та інструментів моделювання онтологій. Особлива увага приділяється аналізу застосування онтологій у різних предметних галузях, де потрібна висока точність прогнозування, з подальшим дослідженням методів оптимізації запитів до онтологічних баз даних. Практична частина роботи передбачає створення демонстраційної онтологічної бази знань для інтелектуальної системи управління віртуальними розподіленими електростанціями, що дасть змогу оцінити практичну цінність дослідження. **Методи дослідження:** ґрунтовний аналіз літератури та доступної документації з теми; порівняльний аналіз досягнутих результатів; подальша демонстрація на практиці застосування обраних інструментів та методів на основі їх показників. **Результати.** Розглянуто питання доцільності онтологічного інжинірингу в сучасних інтелектуальних системах. Наведено та проаналізовано інструменти та методи створення баз знань на основі онтологій. У роботі виконано порівняльну статистику сучасних інструментаріїв зі створення онтологій. Висунуто обґрунтоване бачення ситуації довкола інструментарію та методів онтологічного інжинірингу. Визначено обмеження та унікальні аспекти кожного з розглянутих інструментів. Виокремлено оптимальні підходи до створення баз знань для інтелектуальних систем. На практиці продемонстровано використання та особливості інструментарію онтологічного інжинірингу в контексті інтелектуальних систем із керування віртуальними розподіленими електростанціями. **Висновок.** Новизна дослідження полягає в сучасному погляді на питання вибору інструментів для створення баз знань за допомогою онтологічного інжинірингу, зокрема щодо їх використання в інтелектуальних системах, у поєднанні зі штучним інтелектом. Результати цієї роботи можуть бути впроваджені під час розроблення відповідних інтелектуальних систем, що застосовують бази знань. Дослідження надає широкий спектр інформації та порівнянь підходів та інструментаріїв онтологічного інжинірингу.

Ключові слова: онтологічний інжиніринг; семантичне моделювання; бази знань; онтологічні мови; інструментарій онтологій.

Вступ

Розвиток ІТ-технологій, штучного інтелекту (ШІ) та інтелектуальних систем, побудованих на його основі, є регулярним явищем. Сервіси, що пропонують хмарні потужності ШІ, охоплюють безліч сфер, починаючи від створення зображень або анімованих кліпів (*Dall-E*, *Flux*, *Stable diffusion* тощо), текстових моделей, що пропонують подібно до людини оброблення запитів і навіть написання програмного коду за бажанням користувача, завершуючи інструментарієм зі створення музики. Варто зазначити, що розвиток технологій також досяг того рівня

в питанні продуктивності користувацьких систем, що персональні комп'ютери останніх років так само спроможні відтворювати всі ці можливості локально, без застосування великих серверних потужностей.

Необхідно наголосити на проблематиці нових систем штучного інтелекту. Сучасні моделі ШІ не є подібними до можливостей справжнього мозку людини. Усі згадані вище сервіси та системи є лише результатом опрацювання математично базованим алгоритмом, що має уяву лише про певну предметну галузь. Крім того, що більша система, то вірогідніші помилки та неточності через обмеженість та синтетичність розумової діяльності штучного інтелекту.

Отже, для предметних галузей, що не допускають значних відхилень у результатах роботи штучного інтелекту, необхідне зменшене подання даних для аналізу разом із дуже точно описаними зв'язками та залежностями між елементами. Це зумовлено високою ціною помилкового результату інтелектуальної системи, як, наприклад, у медицині чи в сервісі з віртуально розподіленими електростанціями.

Для подібного уточнення даних для штучного інтелекту можна використовувати підхід онтологічного інжинірингу. Онтологія – це сукупність елементів предметної галузі, їх взаємозв'язків та функцій інтерпретації елементів або зв'язків. Бази знань, побудовані за цим підходом, є чітко визначеними, що знижує ймовірність помилкових результатів роботи ШІ.

Існує чимало мов та інструментів для онтологічного інжинірингу, що мають переваги й недоліки. Нагальним є питання пошуку та підбору інструментарію за його доцільністю.

Основна мета дослідження – визначення сучасних інструментів, встановлення їх параметрів за єдиною системою оцінювання, аналіз можливостей кожного інструменту та демонстрація структурованого результату.

Новизна роботи полягає в аналізі та пошуку найкращого інструментарію та методів онтологічного інжинірингу в контексті інтелектуальних систем на основі порівняння, опрацювання наукових робіт і відкритої інформації щодо кожного ресурсу.

Аналіз предметної галузі

Онтологічний інжиніринг, що впливає з назви, – це сфера знань, яка вивчає сукупність методів і методологій створення баз знань – онтологій [1]. Крім цього, варто надати визначення онтології. У контексті інформаційних технологій та, зокрема, інтелектуальних систем – це база знань, що описує предметну галузь через множину термінів, зв'язків між ними та інтерпретаціями всього зазначеного [2]. Онтології – це формальні семантичні моделі – понятійні бази, що можуть бути прочитані та опрацьовані машинами.

У межах сучасних потреб та запитів ІТ-індустрії наперед виходять інтелектуальні системи із використанням штучного інтелекту. За останні роки такі сервіси були розроблені для багатьох сфер – від творчих галузей, таких як створення зображень, відео, музики, людиноподібних асистентів до більш

серйозних систем, що можуть відповідати за безпеку людини тощо, як, наприклад, автопілот автівки, що платить високу ціну за помилку. Зі свого боку ШІ на поточний момент свого розвитку є лише алгоритмом певного розміру, що лише зчитує запит людини, обробляє його шарами штучних нейронів, після чого формує математично обчислений результат. Отримана відповідь від ШІ може містити похибки чи взагалі помилкові результати тощо. Це обґрунтовано обмеженням навчальних даних, простотою шарів нейронів моделі та найважливіше – ШІ, що сформував власні асоціації між елементами неосязної інформації та видає лише математично прораховану відповідь на основі ймовірностей тощо. Такий підхід є досить доречним у генерації тексту чи медіаконтенту, однак неприпустимий у контексті інтелектуальних систем, що потребують максимально точного результату, як, наприклад, медицина.

Для розв'язання окресленої проблеми можуть бути застосовані онтології як бази знань понятійного апарату штучного інтелекту. У разі використання онтологічного інжинірингу алгоритм буде мати готові, заздалегідь прописані та продумані зв'язки, терміни та їх інтерпретації в предметній галузі. Після застосування баз знань, замість сирієї інформації навчальних даних, ШІ має вираховувати свої відповіді на основі ґрунтовної бази асоціацій та зв'язків.

Питання, зокрема мета цього дослідження, полягає у вивченні, класифікації та аналізі сучасних методів і методологій онтологічного інжинірингу для створення баз знань та подальшого їх використання в інтелектуальних системах, що ґрунтуються на моделях штучного інтелекту та мають високу ціну помилки, як, наприклад, сфера віртуальних розподілених електростанцій.

Для створення формального систематизованого подання галузей знань використовуються мови онтологій, зокрема *RDF (Resource Description Framework)*, *RDFS (Resource Description Framework Schema)*, *OWL (Ontology Web Language)* або їх похідні. Для створення бази знань упроваджується логіка опису. Як уже згадувалось вище, онтологія містить терміни, зв'язки та інтерпретації. У логіці опису інакше це можна назвати класами, ролями та об'єктами відповідно. У галузі комп'ютерних наук існують терміни для цих складників баз знань, а саме *TBox*, *RBox*, *ABox* [3], відповідно до порядку, згаданому вище. Для збереження даних після процесу формалізації аксіом за допомогою логіки опису

можна використовувати *RDF/XML* або *Turtle* тощо. Додатково може бути застосовано *SWRL (Semantic Web Rule Language)* для більшого спектра можливостей логіки опису та правил тощо. Зокрема необхідно зазначити, що, крім звичайних термінів, за допомогою мови *RDF* можна створювати зв'язки з іншою інформацією, такою як зображення, відео тощо. Зі свого боку це розширює межі онтології та дає змогу створювати великі та ґрунтовні семантичні мережі на основі неоднозначних даних [4, 5].

Як вже було зазначено, існує понад одна мова для опису баз знань – онтологій, тож перед їх подальшим і більш детальним розглядом необхідно звернути увагу на *Common Logic*. За своєю сутністю *Common Logic* – це фреймворк для сімейства логічних мов, що оснований на логіці першого порядку та слугує спрощенням обміну даними в комп'ютерних системах. Зокрема це відповідний стандарт ISO/IEC 24707:2018 [6]. *Common Logic* – основа для мов онтологічного інжинірингу, що є його діалектами та отримують від нього абстрактну семантику, яка ґрунтується на теоретичній інтерпретаційній моделі, та синтаксис. Зазначений стандарт виокремлює три напрями діалектів, а саме *Common Logic Interchange Format (CLIF) (Annex A)*, *Conceptual Graph Interchange Format (CGIF) (Annex B)* та оснований на нотації *XML Common Logic (XCL) (Annex C)*. Можна згадати відкритий репозиторій онтологій, створених із використанням *Common Logic* [7], *Hets (The heterogeneous tool set)* [8], і бібліотеку для *PROLOG* мови – *cltools*, що реалізує часткову підтримку *Common Logic* [9] тощо.

Повертаючись до мов онтологічного інжинірингу, хочемо виокремити такі з них: *CycL* – мова проєкту *Cyc* [10]; *Gellish* – онтологічна мова, спрямована на збереження та обмін інформацією [11]; *IDEF5 (Integrated Definition for Ontology Description Capture Method)* – представник сімейства мов моделювання *IDEF* [12]; *KIF* – фреймворк, що більш сконцентрований не на обробленні інформації, а на її передачі між базами знань [13]; *OWL* – нащадок *RDF* та *RDFS*, є мовою онтологічного інжинірингу, спроектованою для використання у *WWW* [14]; *OntoUML* – мова онтологічного інжинірингу, основана на *UML*, що впливає з назви [15]; *SHACL (RDF SHapes Constraints Language)* – стандартна мова *W3C*, яка підтримує також

взаємодію із *OWL* та *RDFS* [16]; *XBRL (Extensible Business Reporting Language)* – фреймворк для створення онтологій, але більш зосереджений на бізнес-складнику [17]; *SPARQL (SPARQL Protocol and RDF Query Language)* – мова запитів до даних за моделлю *RDF* [18].

Розглядаючи мови та фреймворки онтологічного інжинірингу, необхідно обрати інструментарії та програми, що використовують відповідні мови для створення баз знань. Найбільш уживаними можна назвати такі: *Hozo* [19], *Lettria's Ontology Toolkit* [20], *Protégé* [21], *Onto4ALL* [22], *FluentEditor* [23]. Перелічені застосунки є інструментами для створення редакції та подальшого впровадження онтологій, однак вони розрізняються за функціоналом та можливостями. Порівняння їх властивостей подано в табл. 1.

У контексті розгляду питання про вибір кращого інструментарію для онтологічного інжинірингу для інтелектуальних систем є доречним також порівняти мови, що використовуються в застосунках, які розглядаються. Порівняння подано в табл. 2.

З наведених порівнянь застосунків-інструментаріїв та мов онтологічного інжинірингу (табл. 1 та 2 відповідно), можна прийти до обґрунтованих висновків. Кожен із застосунків підходить найбільше під власні спеціалізовані завдання. Так, для складних онтологій без текстової інформації найбільш підходить *Protégé*. Зі свого боку *Lettria's Ontology Toolkit* пропонує дуже потужний функціонал для оброблення текстових даних та створення баз знань на їх основі, що зосереджує його на текстових моделях штучного інтелекту, зокрема *ChatGPT*, *Gemini* тощо. *FluentEditor* має менш сфокусовану сферу застосування, однак завдяки підтримці *CNL* створення за його допомогою онтологій є більш інтуїтивним. *Onto4ALL* та *Hozo* є більш простими застосунками із загальним спектром можливостей, особливо впроваджуються в незначних завданнях і проєктах.

Однак з огляду на фактори доступності, зручності та можливостей найбільш бажаними інструментаріями є *Protégé* – для більш класичних онтологій, побудованих на термінах, відношеннях та інтерпретаціях, а для великих складних проєктів, що будуються навколо текстових баз знань, підійде *Lettria's Ontology Toolkit*, особливо зважаючи на її *enterprise*-формат поширення.

Таблиця 1. Порівняльна характеристика інструментів онтологічного інжинірингу

Критерій	Hozo	Lettria's Ontology Toolkit	Protégé	Onto4ALL	FluentEditor
Графічний інтерфейс	Певною мірою застарілий, проте зрозумілий; достатній для роботи, але вимагає покращень.	Інтуїтивний, сучасний, зосередженість на штучному інтелекті; гарна візуалізація.	Відносно сучасний, зрозумілий, багатофункціональний.	Інтуїтивний, сучасний, зручний, достатньо інформативний.	Певною мірою застарілий, менш функціональний та інформативний.
Підтримка мов, форматів та фреймворків	Text/HTML, CSV, Hozo Text, XML-DTD, DAML+OIL, RDF, OWL (A, B, C, Old), Markdown, RDF (Relation), Sub-Language Data.	OWL, JSON-LD, RDF.	RDF/XML, Turtle, OWL/XML, OWL Functional, Manchester OWL, OBO, LaTeX, JSON-LD, SWRL, SPARQL, SQWRL. With plugins: Excel workbook, OAF, ROWL, SHACL.	OWL, OWL/XML, OWL, XML, TURTLE, SINTAXEMNCHESTER, RDF/XML, SVG.	CNL, RDF/XML, OWL/XML, HTM/HTML, SKOS, SPARQL.
Особливість інструментарію	Простий функціональний застосунок із достатніми можливостями; управління залежностями, колаборативна робота, API для інтеграції.	Генерація онтологій з текстів, аналіз на основі NLP; висока сумісність вихідного формату, зручність та простота.	Багатофункціональний застосунок із можливістю встановлювати плагіни, що розширюють можливості ще більше.	Простий вебзастосунок з інтуїтивним інтерфейсом та функціоналом.	Підтримка CNL, зручна робота із семантичними конструкціями, багатоінформативний інтерфейс.
Можливості співпраці	Підтримує спільне розроблення з менеджером версій.	Обмежена, зосередженість на автономне редагування.	Розширювана через плагіни та інструменти для командної роботи.	Не підтримує.	Підтримує інтеграцію з різними редакторами та онлайн-співпрацю.
Простота використання	Середня, вимагає базового навчання.	Висока, завдяки AI-функціям та інтуїтивному інтерфейсу.	Складна для новачків, але потужна для експертів.	Простий у використанні, зосередженість на початківцях або на маленьких проєктах.	Середня завдяки природній мові та змістовному інтерфейсу, однак вимагає навчання.
Доступність	Desktop-застосунок із вбудованим сервером, у відкритому доступі.	Вебзастосунок, однак поширюється за enterprise-форматом.	Desktop-застосунок у відкритому доступі.	Вебзастосунок у відкритому доступі.	Desktop-застосунок у відкритому доступі, зокрема в Microsoft Store.

Таблиця 2. Порівняльна характеристика мов онтологічного інжинірингу

Мова	Характеристики	Переваги	Недоліки	Основне застосування
1	2	3	4	5
Text/HTML	Простий текст або гіпертекстовий формат.	Легкий для створення та читання, підходить для невеликих онтологій.	Відсутність структурованості та семантичних зв'язків.	Документація, опис простих моделей.
CSV	Табличний формат, поділений комами.	Зручний для табличних даних, легко обробляється програмно.	Складний для моделювання ієрархій і складних зв'язків.	Таблиці даних, прості експорт / імпорт.
XML-DTD	Визначення структури XML-документів, що задає допустимі елементи та атрибути.	Чіткість та структурованість, стандарт у XML.	Старий формат, складний для оброблення сучасними інструментами.	Валідація XML-документів.

Продовження таблиці 2

1	2	3	4	5
<i>DAML+OIL</i>	Попередник <i>OWL</i> , розроблений для моделювання онтологій в <i>Semantic Web</i> .	Семантична потужність, гнучкість у створенні складних моделей.	Обмежена підтримка сучасними системами.	Моделювання семантичних систем (здебільшого застарілий).
<i>RDF</i>	Модель даних, що зберігає інформацію у вигляді триплетів (суб'єкт – предикат – об'єкт).	Простота, широко використовується в <i>Semantic Web</i> .	Відсутність підтримки складних аксіом або обмежень.	Семантичні вебресурси, базові моделі онтологій.
<i>OWL</i>	Високорівнева мова для створення складних онтологій, містить варіанти <i>OWL (A, B, C)</i> .	Підтримка складних зв'язків, сумісність із <i>RDF</i> .	Вимагає вивчення, складність реалізації.	Комплексні онтології, опис знань.
<i>OWL/XML, RDF/XML</i>	<i>XML</i> -варіанти <i>OWL</i> і <i>RDF</i> , що додають чіткість та зручність оброблення.	Уніфікація із <i>XML</i> -системами, підтримка багатьма інструментами.	Перевантаженість синтаксису, складність читання вручну.	Інтеграція з <i>XML</i> -системами.
<i>Turtle</i>	Спрощений синтаксис для <i>RDF</i> .	Зручність для розробників, компактний і легко читається.	Лімітований для складних структур.	Розроблення семантичних вебресурсів.
<i>JSON-LD</i>	Легковагова структура для опису даних у <i>JSON</i> з додаванням семантики.	Простота інтеграції з вебтехнологіями, легкість у розробленні.	Обмежена семантична потужність порівняно з <i>OWL</i> .	Вебзастосунки, інтеграція з <i>API</i> .
<i>SWRL</i>	Мова правил для <i>OWL</i> .	Розширення можливостей <i>OWL</i> , дає змогу створювати логічні правила.	Складність у реалізації та виконанні.	Додавання правил у складні онтології.
<i>SPARQL</i>	Мова запитів для <i>RDF</i> -даних.	Потужна для запитів семантичних даних.	Використовується лише для читання, не модифікації.	Пошук і оброблення семантичних даних.
<i>SKOS</i>	Простий об'єктний стандарт для моделювання класифікацій і словників.	Простота використання, підтримка семантики.	Обмежені можливості для складних онтологій.	Тезауруси, словники, класифікації.
<i>CNL</i>	Контрольована природна мова для створення й редагування онтологій.	Зрозумілість для людей, простота редагування.	Обмежений набір інструментів для використання.	Гуманізовані онтології, інтеграція з текстовими інтерфейсами.

Також варто розглянути, які саме мови та фреймворки є найбільш актуальними для онтологічного інжинірингу в контексті створення інтелектуальних систем. Мови *RDF* і *OWL* все ще залишаються стандартами для великих онтологій з підтримкою складних зв'язків і можливістю інтеграції з іншими форматами. *Turtle* і *JSON-LD* ефективні для швидкого розроблення та інтеграції з вебтехнологіями завдяки простому синтаксису. Мова запитів *SPARQL* надає потужні засоби для запитів семантичних даних, що робить її необхідним інструментом для аналізу онтологій.

Аналіз наукових досліджень у галузі

Як зазначалося вище, сучасний етап розвитку інформаційних технологій, зокрема в сферах штучного інтелекту та семантичного вебу, диктує необхідність постійного вдосконалення методів та інструментів онтологічного інжинірингу. Навіть за наявності добре усталених підходів, таких як *RDF*, *OWL* та *SPARQL*,

науковці не припиняють пошук інноваційних рішень для підвищення ефективності процесів створення, управління та практичного застосування онтологій. За останні п'ять років спостерігаємо значне дослідницьке зацікавлення до інтеграції онтологій з методами машинного навчання та ШІ. Цей тренд спрямований на створення більш адаптивних, інтелектуальних і автономних систем.

Одним із ключових напрямів сучасних досліджень є розроблення методів автоматизованого створення та динамічного поповнення онтологій. Традиційний процес онтологічного інжинірингу, як правило, є досить ресурсоемним і вимагає значної участі експертів. Утім сучасні підходи дають змогу спростити та оптимізувати цей процес. У дослідженні [24] пропонується нова методологія, що використовує глибокі нейронні мережі для вилучення онтологічних концепцій та семантичних відношень безпосередньо із текстових даних. Цей підхід оснований на архітектурі трансформерів, що дає змогу ефективно

аналізувати контекст і виявляти ключові елементи онтологій. Застосування методів глибокого навчання сприяє автоматизації рутинних етапів, що значно прискорює створення нових онтологій та підвищує їх якість. Крім того, нейронні мережі здатні розпізнавати неявні семантичні зв'язки між різними термінами, що не завжди є очевидним для експерта. У статті [25] запропоновано інтегрований метод, що поєднує лінгвістичний аналіз, зокрема оброблення природної мови (*NLP*), та семантичні мережі для автоматичного виявлення та структурування онтологічних знань із великих масивів неструктурованих даних. Цей метод використовує комбінацію алгоритмів розпізнавання іменованих сутностей, семантичного розпізнавання ролей та інших *NLP*-методів для ідентифікації онтологічних класів, відношень та атрибутів. Особлива увага приділяється обробленню неоднозначних термінів за допомогою *WordNet* та їх інтеграції в структуру онтології, що сприяє створенню більш точних і надійних онтологій. Важливими є належне оброблення неоднозначних термінів і контекстуалізація виявлених фактів, що сприяє створенню більш точних та надійних онтологій із збереженням часової ефективності.

Крім того, дослідників цікавить створення інструментів для візуалізації онтологій, які б відповідали викликам сучасних інтелектуальних систем. У роботі [26] описано інтерактивний інструмент *PRONTOVISE*, що ґрунтується на концепціях когнітивних карт для візуалізації знань і семантичних зв'язків. Цей інструмент надає розширені можливості для подання складних онтологічних просторів, спрощує процес вивчення онтологій та полегшує їх розуміння, що є вкрай важливим для моделювання інтелектуальних систем реального часу. Когнітивні карти дають змогу подавати знання в більш інтуїтивно зрозумілому вигляді, що спрощує процес моделювання та полегшує міждисциплінарну співпрацю між експертами з різних предметних галузей. У статті [27] запропоновано новий інструмент для інтерактивної 3D-візуалізації онтологій, що допомагає користувачам досліджувати ієрархії класів та імпортовані терміни в тривимірному просторі. Інструмент сприяє кращому розумінню складних онтологічних структур завдяки зручній навігації та інтерактивним графічним елементам.

Варто також зазначити, що в дослідженнях спостерігається тенденція до розширення

функціональності наявних мов, наприклад *OWL*, способом додавання нових конструкцій та правил [28–31]. Це дає змогу більш точно та детально описувати знання, зважаючи на особливості конкретних предметних галузей. Зокрема пропонуються нові методи для опису часових залежностей між різними поняттями або для уважного ставлення до контекстуальної інформації в процесі виведення нових фактів.

Іншим важливим аспектом є дослідження методів ефективного використання онтологій у різних предметних галузях. У роботах [32–34] детально проаналізовано застосування онтологічного інжинірингу для створення інтелектуальних систем у сфері охорони здоров'я, зокрема для розроблення систем підтримки прийняття клінічних рішень та інтеграції медичних знань із різних джерел. Автори наголошують, що онтології забезпечують формальне та структуроване подання медичних даних, що сприяє більш ефективному ухваленню клінічних рішень і підвищенню якості медичного обслуговування. У праці [35] розглядається застосування онтологій у сфері енергетики, зокрема для моделювання розумних енергетичних мереж, аналізу електроенергетичних ринків та інтеграції різних систем управління енергоресурсами. Автори статті [36] досліджують застосування онтологій у системах управління ланцюгами постачання, де онтології впроваджуються для моделювання та відстеження логістичних процесів, забезпечуючи прозорість і покращену взаємодію між учасниками ланцюга постачання.

Щодо підвищення продуктивності онтологічного інжинірингу існують дослідження, присвячені оптимізації запитів до онтологій та підвищенню швидкості оброблення семантичних даних. У роботі [37] запропоновано інноваційну структуру індексації для *RDF*-даних, що дає змогу значно прискорити виконання *SPARQL*-запитів, зменшуючи час пошуку необхідної інформації та підвищуючи масштабованість онтологічних систем. Також спостерігається тенденція щодо використання паралельних обчислень та розподілених систем для оброблення великих масивів онтологічних даних [38–40]. У статті [41] описано ключові напрями використання онтологій у хмарних обчисленнях, зокрема для забезпечення інтероперабельності, управління ресурсами та вибору сервісів. Обговорюються методи масштабування для

ефективного керування значними обсягами даних, що дає змогу використовувати онтології в обробленні великих даних та багатомарних середовищах.

Аналіз сучасних досліджень у сфері онтологічного інжинірингу свідчить про активний розвиток методів, інструментів та мов для створення, управління та застосування онтологій. Зокрема спостерігається тенденція до автоматизації процесів онтологічного інжинірингу, розширення можливостей наявних мов та інструментів, а також пошуку нових підходів до застосування онтологій у різних предметних галузях. Результати цих досліджень доводять важливість онтологій для створення інтелектуальних систем, що є особливо актуальним у контексті зростання складності сучасних інформаційних технологій.

Результати практичного дослідження

На основі проаналізованої інформації та зроблених проміжних висновків розглянемо результати їх застосування в онтологічному інжинірингу бази знань для інтелектуальної системи. Для зразкової предметної галузі було обрано тематику застосунку віртуальних розподілених електростанцій. У цій інтелектуальній системі передбачено на основі якісних онтологій та їх використання разом із штучним інтелектом запровадити застосунок, що розв'язує проблему розподілу ресурсів за наявними вимогами та потужностями [42–45].

Як зазначалося вище, галузь віртуальних розподілених систем має високу ціну помилкового прогнозування штучним інтелектом, саме тому для розв'язання окресленої проблеми доречно застосовувати онтологічний інжиніринг для надання більш точного контексту інтелектуальній системі, зокрема III.

Отже, метою практичної частини цього дослідження є демонстрація ефективності інструментарію, який розглянуто вище, та навести порівняння із спорідненими підходами до виконання обраного завдання (створення інтелектуальної системи – застосунку для керування віртуальними розподіленими електростанціями), що має досить високу потребу в правильних прогнозах штучного інтелекту.

Як інструментарій було обрано оптимальне поєднання – застосунок *Protégé* та мову запитів *SPARQL* для взаємодії із наповненням бази знань.

Також додатково буде застосовано *SWRL* з метою створення логічних правил у онтології [46–48].

Необхідно зазначити, що в межах демонстративної практичної частини буде створено онтологію лише за незначною частиною предметної галузі інтелектуальної системи контролю за віртуальними розподіленими електростанціями. Цього обсягу даних має бути достатньо для аналізу та ґрунтового висновку щодо використання онтологічного інжинірингу для подібних завдань.

Отже, розглянемо структуру та будову спрощеної онтології у формалізованому вигляді [49].

1. Класи та підкласи:

а) Generator – джерело енергії:

- SolarPanel – сонячна панель;
- WindTurbine – вітряна турбіна;
- HydroGenerator – гідроелектростанція;

б) Consumer – споживач енергії:

- ResidentialConsumer – житловий споживач;
- IndustrialConsumer – промисловий споживач;

в) EnergyNode: вузли розподілу енергії:

- MainNode – основний вузол;
- SubNode – підпорядкований вузол;

г) Contract: угода між генератором і споживачем

із параметрами:

- EnergyAmount – кількість енергії;
- StartDate, EndDate – дати дії контракту.

2. Полі (зв'язки):

а) providesEnergy: зв'язок між Generator і EnergyNode;

б) consumesEnergy: зв'язок між Consumer і EnergyNode;

в) hasContract: зв'язок між Generator, Consumer і Contract.

Зокрема необхідно виокремити "суперкласи", або загальні концепти, такі як *Entity*, *DistributionPoint* і *Agreement*.

Для більшої ґрунтовності та наочності зв'язків і будови створюваної бази знань – онтології – варто розглянути її з погляду логіки опису – математичної моделі. Вона формалізує відношення між класами, підкласами, індивідами та їх зв'язками.

Відповідно до термінологічної бази або *TBox* нижче подамо згадану онтологію.

1) Класи та підкласи:

а) Generator $\sqsubseteq$ Entity: джерело енергії є сутністю;

• SolarPanel $\sqsubseteq$ Generator: сонячні панелі є генераторами;

• WindTurbine $\sqsubseteq$ Generator: вітряні турбіни є генераторами;

• HydroGenerator $\sqsubseteq$ Generator:
гідроелектростанції є генераторами;

б) Consumer $\sqsubseteq$ Entity: споживач енергії є сутністю;

• ResidentialConsumer $\sqsubseteq$ Consumer: житлові споживачі є споживачами;

• IndustrialConsumer $\sqsubseteq$ Consumer:
промислові споживачі є споживачами;

в) EnergyNode $\sqsubseteq$ DistributionPoint: вузли енергорозподілу є точками розподілу;

• MainNode $\sqsubseteq$ EnergyNode: основний вузол є вузлом розподілу енергії;

• SubNode $\sqsubseteq$ EnergyNode: підпорядкований вузол є вузлом розподілу енергії;

г) Contract $\sqsubseteq$ Agreement: контракти є угодами;

• Contract $\sqsubseteq \exists \text{hasAttributes.}(\text{EnergyAmount} \sqcap \text{StartDate} \sqcap \text{EndDate})$: контракт містить параметри кількості енергії та дат початку й завершення.

2) Відношення (ролі):

а) ProvidesEnergy $\sqsubseteq$ Relation: генератори постачають енергію до вузлів;

• Generator $\sqsubseteq \exists \text{providesEnergy.EnergyNode}$: кожен генератор постачає енергію до вузла;

б) ConsumesEnergy $\sqsubseteq$ Relation: споживачі отримують енергію з вузлів;

• Consumer $\sqsubseteq \exists \text{consumesEnergy.EnergyNode}$: кожен споживач отримує енергію з вузла;

в) hasContract $\sqsubseteq$ Relation: зв'язок між генератором, споживачем і контрактом;

• Generator $\sqsubseteq \exists \text{hasContract.Contract}$: генератори укладають контракти;

• Consumer $\sqsubseteq \exists \text{hasContract.Contract}$: споживачі мають контракти.

Розглядаючи базу знань під кутом *RBox* або властивостей та обмежень ролей, подамо такий опис:

• функціональність – $\forall \text{providesEnergy.} \neg \text{providesEnergy}^{\wedge}$: генератор може постачати енергію лише одному вузлу одночасно;

• унікальність контрактів – $\forall \text{hasContract.}(\exists \text{linkedByContract.Generator})$: споживач може мати контракт лише з одним генератором;

• обмеження вузлів – $\forall \text{providesEnergy.}(\leq \text{MaxCapacity.EnergyNode})$: загальна кількість енергії, що постачається генераторами, не має перевищувати потужності вузла.

Зі свого боку асерційна база *ABox* представляє індивіди та відповідні зв'язки між ними для онтології, що створюється. Необхідно зазначити, що ця онтологія є лише демонстративним прикладом – наочним результатом дослідження, тож дані є лише схематичним нарисом та здебільшого заглушками й макетами, що не відтворюють повною мірою потенційно справжній вміст такої онтології на практиці.

Індивіди:

- SolarPanel_1 $\sqsubseteq$ SolarPanel;
- WindTurbine_2 $\sqsubseteq$ WindTurbine;
- Factory_1 $\sqsubseteq$ IndustrialConsumer;
- Home_1 $\sqsubseteq$ ResidentialConsumer;
- MainNode_1 $\sqsubseteq$ MainNode;
- Contract_101 $\sqsubseteq$ Contract.

Зв'язки між індивідами:

• providesEnergy(SolarPanel_1, MainNode_1): сонячна панель 1 постачає енергію до основного вузла 1;

• consumesEnergy(Factory_1, MainNode_1):

завод 1 отримує енергію з основного вузла 1;

• hasContract(SolarPanel_1, Contract_101):

контракт 101 пов'язаний із сонячною панеллю 1;

• hasContract(Factory_1, Contract_101):

контракт 101 пов'язаний із заводом 1.

Також варто окремо подати математичні аксіоми онтології, що розглядається.

1) Генератор, споживач та вузли:

- Generator $\sqsubseteq$ Entity;
- Consumer $\sqsubseteq$ Entity;
- EnergyNode $\sqsubseteq$ DistributionPoint.

2) Взаємозв'язки:

- providesEnergy(SolarPanel_1, MainNode_1)

$\Rightarrow$ SolarPanel_1 $\sqsubseteq \exists \text{providesEnergy.MainNode}$;

- consumesEnergy(Factory_1, MainNode_1) $\Rightarrow$

Factory_1 $\sqsubseteq \exists \text{consumesEnergy.MainNode}$.

3) Контракти:

- Contract_101 $\sqsubseteq$

$\exists \text{hasAttributes.}(\text{EnergyAmount} \sqcap \text{StartDate} \sqcap \text{EndDate})$;

- hasContract(SolarPanel_1, Contract_101) $\sqcap$

hasContract(Factory_1, Contract_101) $\Rightarrow$

$\exists \text{linkedByContract.Generator}$.

Завдяки такому змістовному опису забезпечується формальна структура онтології в предметній галузі, що дає змогу виконувати логічне виведення для оптимізації роботи інтелектуальної системи з контролю віртуальних розподілених електростанцій. Логіка опису гарантує точність та стійкість

моделювання, гнучкість і можливість легкого масштабування застосунку.

Як вже зазначалося раніше, для онтологічного інжинірингу бази знань у контексті розподілених віртуальних електростанцій доречно використати *SPARQL*. Інтеграція мови запитів до бази знань дає змогу ефективно отримувати, аналізувати й модифікувати дані, зберігаючи їх формалізовану структуру й семантику, аналогічно до мови *SQL* тощо.

Необхідно відразу виокремити декілька важливих запитів, що мають бути додані до онтологічної моделі, що розробляється.

По-перше, код запиту на отримання всіх споживачів від конкретного вузла; це може бути корисним у багатьох ситуаціях використання інтелектуальної системи:

```
1. SELECT ?consumer
2. WHERE {
3.     ?consumer :consumesEnergy
      :MainNode_1.
4. }
```

Іншим, проте не менш корисним, є запит, що шукає перенавантажені вузли та допомагає в підтримці всієї системи бути стабільною, незважаючи на коливання потреб користувачів та наявних потужностей:

```
1. SELECT ?generator ?node
   (SUM(?energyAmount) AS ?totalEnergy)
2. WHERE {
3.     ?generator :providesEnergy ?node.
4.     ?generator :energyAmount
      ?energyAmount.
5. }
6. GROUP BY ?node
```

```
7. HAVING (?totalEnergy > ?nodeCapacity)
```

Запит, що відповідає за пошук активних користувачів, із актуальними контрактами на отримання потужностей системи розподілених станцій:

```
1. SELECT ?generator ?consumer
   ?energyAmount ?startDate ?endDate
2. WHERE {
3.     ?contract :linkedByContract
      ?generator.
4.     ?contract :linkedByContract
      ?consumer.
5.     ?contract :energyAmount
      ?energyAmount.
6.     ?contract :startDate ?startDate.
7.     ?contract :endDate ?endDate.
8.     FILTER (NOW() >= ?startDate &&
      NOW() <= ?endDate)
9. }
```

Завдяки мові запитів *SPARQL* та, зокрема, запитам на ній є можливість машинної швидкої взаємодії із відповідними даними. Зважаючи на це, онтологія, поєднуючись із цією мовою, підтримує оброблення штучним інтелектом або будь-яким іншим алгоритмом для аналізу даних та подальшого прогнозування й автоматизації процесів, що необхідно в обраному контексті – предметній галузі програмного застосунку – інтелектуальній системі з керування віртуальними розподіленими електростанціями.

Також варто порівняти підходи зі звичайним збереженням даних, наприклад, у реляційних базах даних із використанням мови запитів *SQL* та онтологіях у форматі триплетів "суб'єкт – предикат – об'єкт" із застосуванням *SPARQL*. Порівняльна характеристика наведена в табл. 3.

Таблиця 3. Порівняльна характеристика *SPARQL* (онтології) та *SQL* (реляційні бази даних)

Критерій 1	SPARQL (онтології) 2	SQL (реляційні бази даних) 3
Модель даних	Ґрунтується на <i>RDF</i> -триплетях (суб'єкт – предикат – об'єкт), що дає змогу моделювати семантичні зв'язки між сутностями.	Використовує реляційну модель, основу на таблицях, де дані зберігаються у вигляді рядків і стовпців.
Типи запитів	Орієнтований на графові дані; уможливує здійснення запитів до даних, пов'язаних семантичними зв'язками.	Орієнтований на табличні дані; добре підходить для агрегованих, транзакційних запитів.
Підтримка семантики	Підтримує семантичні відношення між даними; дає змогу здійснювати запити з огляду на логічні правила й аксіоми онтології.	Не має вбудованої семантики; зв'язки між таблицями задаються вручну за допомогою ключів (<i>primary/foreign keys</i>).
Масштабованість	Висока для структур із складними зв'язками (онтології, графи). Ефективний для динамічних, розподілених структур.	Краще працює для великих транзакційних даних із простою структурою (таблиці, індекси).
Адаптація до змін	Гнучка структура <i>RDF</i> дає змогу легко додавати нові сутності, відношення чи властивості без порушення наявних даних.	Зміна структури (додавання стовпців чи таблиць) може вимагати міграції даних та складного налаштування.

Продовження таблиці 3.

1	2	3
Підтримка інтеграції	Призначений для роботи з онтологіями та <i>Semantic Web</i> ; дає змогу інтегрувати різні джерела даних за допомогою стандартів <i>RDF/OWL</i> .	Вимагає складнішої інтеграції між різними системами; немає вбудованої підтримки <i>Semantic Web</i> .
Логічні виведення	Дає змогу здійснювати запити з огляду на логічні виведення (<i>Reasoning</i>), наприклад, за допомогою <i>SWRL</i> або інших інструментів онтологій.	Логічні виведення не підтримуються; потрібно реалізовувати бізнес-логіку окремо на стороні застосунку.
Продуктивність	Ефективний для роботи з графами та пов'язаними даними, але може бути повільнішим для великих транзакційних даних.	Висока швидкість для оброблення значних обсягів структурованих даних у табличному вигляді.
Інструменти та підтримка	Підтримується інструментами, такими як <i>Protégé</i> , <i>GraphDB</i> , <i>Blazegraph</i> ; інтеграція із <i>Semantic Web</i> .	Широка підтримка СУБД (<i>MySQL</i> , <i>PostgreSQL</i> , <i>Oracle</i> , <i>SQL Server</i>); підходить для традиційних застосунків.
Використання в інтелектуальних системах з керування віртуальними розподіленими електростанціями	Ідеально підходить для моделювання складних взаємозв'язків у системах управління, таких як зв'язки між генераторами, вузлами й контрактами.	Ефективно для зберігання транзакційних даних, таких як журнали споживання чи фінансові операції.

Також зазначимо важливість використання *SWRL* – мови правил для *OWL*. Вона дає змогу описувати логічні правила за умови взаємодії абстракцій – концептів – та є розширенням до базових можливостей онтологічного інжинірингу, зокрема мови *OWL* тощо. У контексті інтелектуальних систем із важливістю правильних прогнозів алгоритмів штучного інтелекту, як тої, що розглядається, – застосунок із керуванням

віртуальними розподіленими електростанціями – *Semantic Web Rule Language* є інструментом для постанови та оптимізації логічних процесів тощо. Узагальнюючи, *SWRL* додає предикатну логіку до відношень та правил за допомогою форми "якщо, то".

Для більшого занурення в питання впровадження *SWRL* варто порівняти можливості звичайної *OWL* та онтологій із використанням мови правил (табл. 4).

Таблиця 4. Порівняльна характеристика базової *OWL* та онтологій із *SWRL*

Критерій	Логіка опису (<i>OWL</i>)	<i>SWRL</i>
Формалізм	Формальні аксіоми	Розширення логіки за допомогою правила "якщо, то". Правила <i>SWRL</i> можуть бути використані для виведення нових фактів, наприклад ідентифікації споживачів, які можуть отримати додаткову енергію за наявності надлишкової потужності.
Можливості	Обмежений набір виразів	Динамічні правила для складних умов. Правила легко адаптуються до змін у системі, наприклад додавання нових типів генераторів або споживачів.
Динамічність	Фіксована структура онтології	Гнучкість для створення адаптивних рішень. Зокрема <i>SWRL</i> дає змогу автоматизувати складні процеси, наприклад перенаправлення енергії від перевантажених вузлів до менш завантажених.
Застосування в штучному інтелекті	Пасивна база знань	Активне виведення нових знань і оптимізація рішень. <i>SWRL</i> може бути використана разом із <i>Reasoner</i> (наприклад, <i>Hermit</i> або <i>Pellet</i>) для розширення можливостей штучного інтелекту, зокрема автоматизації управління та прогнозування.

Зокрема можна виокремити низку правил, що є ефективними для впровадження інтелектуальної системи з контролю віртуальними розподіленими електростанціями:

- правило з управління перенавантаженням вузлів – якщо вузол перевантажений, то перенаправити енергію на інший вузол: $\text{EnergyNode}(?n1) \wedge \text{EnergyNode}(?n2) \wedge$

$\text{Overloaded}(?n1) \wedge \text{Connected}(?n1, ?n2) \rightarrow \text{RedirectEnergy}(?n1, ?n2);$

- правило з оптимізації контрактів – якщо контракт між генератором і споживачем має вичерпаний термін дії, то позначити контракт як неактивний: $\text{Contract}(?c) \wedge \text{hasEndDate}(?c, ?end) \wedge \text{CurrentDate}(?now) \wedge ?end < ?now \rightarrow \text{DeactivateContract}(?c);$

- правило про балансування енергоспоживання – якщо споживач перевищує допустиме споживання, зменшити потужність, яку надає генератор:
 $\text{Consumer}(?c) \wedge \text{consumesEnergy}(?c, ?n) \wedge \text{Overconsuming}(?c) \rightarrow \text{AdjustOutput}(?n, ?c);$

- правило про прогнозування перевантажень – якщо загальне споживання на вузлі перевищує 90 % потужності, позначити вузол як потенційно перевантажений:
 $\text{EnergyNode}(?n) \wedge \text{TotalConsumption}(?n, ?t) \wedge \text{MaxCapacity}(?n, ?m) \wedge (?t > 0.9 * ?m) \rightarrow \text{PotentialOverload}(?n).$

Також варто зазначити, що інструмент *Protégé* має підтримку *Reasoner* та, зокрема, мови *SWRL*.

Також перелічимо, що саме робить *Reasoner*. Він визначає, чи є правила узгодженими з онтологією, чи ні. Зі свого боку правила *SWRL* генерують нові факти, що зберігаються в базі знань для подальшого використання. Також *Reasoner* виконує правила в реальному часі, забезпечуючи автоматичний контроль системи.

Результатом відтворення описаних вище елементів та складників онтології розподілених віртуальних електростанцій стала демонстративна база знань, що зображена на рис. 1.

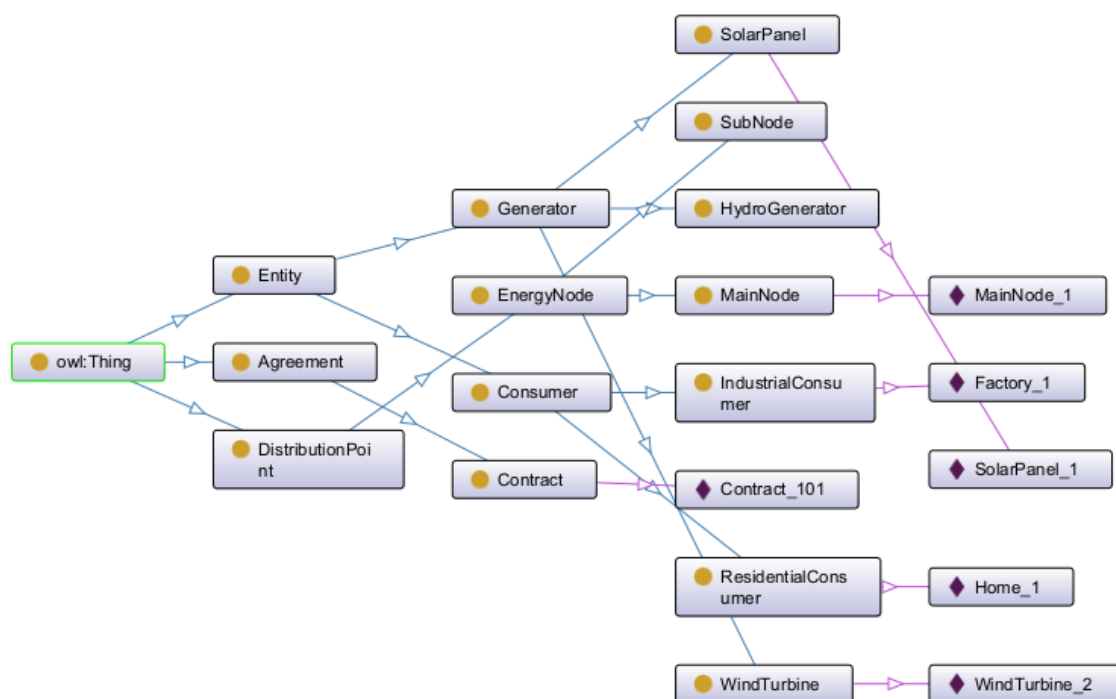


Рис. 1. Демонстративна онтологія за предметною галуззю віртуальних розподілених електростанцій

Отже, як практична демонстрація була розглянута скорочена реалізація онтологічного інжинірингу для інтелектуальної системи управління віртуальними розподіленими електростанціями. Вона підтвердила ефективність запропонованих методів і інструментів у розв'язанні проблеми уточнення відповідей ІІІ. Було розглянуто та оприлюднено процес побудови бази знань із використанням застосунку *Protégé* та мов *OWL*, *SPARQL* і *SWRL*. Показано важливість і доцільність інтеграції онтологій у сучасні інтелектуальні системи, що використовують ІІІІ або інші алгоритми прогнозування відповіді за умови високої ціни помилкового результату.

Підсумовуючи результати практичної частини дослідження, можна наголосити, що онтологічний інжиніринг забезпечує формальну структуру даних, що мінімізує ризики помилок у прогнозах і розрахунках. Використання *RDF*-структур дає змогу легко розширювати базу знань, додаючи нові класи, властивості та правила без порушення наявної моделі. *SPARQL* і *SWRL* забезпечують глибокий аналіз даних, автоматизацію логічних операцій та можливість прогнозування на основі поточного стану системи. Інтелектуальна система на основі онтологій є сумісною з алгоритмами штучного інтелекту, що сприяє розробленню самонавчальних і прогнозуючих моделей.

Висновки

У статті проаналізовано галузь знань онтологічного інжинірингу. Розглянуто важливість застосування онтологій з метою уточнення даних для ІІІ та інших інтелектуальних систем, що математично прогнозують відповіді на основі базової інформації за галуззю знань. Підставою для дослідження стала нагальна потреба в уточненні системи прогнозування відповідей штучним інтелектом, зокрема для галузей, де ціна помилки надто висока, як, наприклад, медицина чи інтелектуальні системи із віртуальними розподіленими електростанціями тощо.

Досліджено наявні мови та фреймворки онтологічного інжинірингу та розглянуто інструментарій, що використовує логіку цих мов для створення баз знань. У статті подано актуальні застосунки, спроектовані для створення та редагування онтологій.

Наведено порівняльні характеристики для обох аспектів онтологічного інжинірингу: як інструментів – застосунків, так і формальних мов опису баз знань. Під час порівняння всебічно розглянуто предмети аналізу, завдяки чому створено обґрунтовану базу знань про інструментарій із можливістю подальшого формулювання висновків на цій основі.

Ця робота відповідає на питання пошуку актуальних методів та інструментів створення баз

знань із використанням онтологічного інжинірингу в контексті інтелектуальних систем. Результатом дослідження стає висунення найбільш оптимальних сценаріїв для використання кожного із застосунків та онтологічних мов, апелюючи до їх параметрів і характеристик, порівнюючи їх з іншими, що розглядалися в роботі.

Дослідження містить практичну демонстрацію та закріплення висновків, які були зроблені під час аналізу наявного інструментарію та мов онтологічного інжинірингу. Зокрема була створена обмежена версія онтології для інтелектуальної системи з керування віртуальними розподіленими системами.

Уся практична частина була основана на закріпленні відомостей про оптимальний інструментарій – *Protégé*, та відповідні мови – *OWL*, *SPARQL*, *SWRL*. Проаналізовано використання онтологічного інжинірингу для створення нарисів предметної галузі машинною мовою порівняно з неонтологічним підходом та, зокрема, переваги мови правил *SWRL* над базовою логікою *OWL* тощо.

Досягнуті результати свідчать про високу практичну цінність онтологічного інжинірингу в умовах, де точність прогнозів і адаптивність системи є критично важливими. Подальший розвиток цього підходу може передбачати розширення онтології, оптимізацію правил та інтеграцію з розподіленими обчислювальними системами для забезпечення масштабованості.

Список літератури

1. Pouchard L., Ivezić N., Schlenoff C. Ontology engineering for distributed collaboration in manufacturing. AIS2000 Conf. 2000. 1012 p. URL: https://www.researchgate.net/publication/228596365_Ontology_engineering_for_distributed_collaboration_in_manufacturing (дата звернення: 02.12.2024).
2. Guarino N. Ontologies and knowledge bases: towards a terminological clarification, Towards Very Large Knowl. Bases. Amsterdam:IOS Press. 1995. P. 25–32. URL: https://www.researchgate.net/publication/220041941_Ontologies_and_knowledge_bases_towards_a_terminological_clarification (дата звернення: 02.12.2024).
3. Motz R., Rohrer E., Severi P. The description logicSHIQwith a flexible meta-modelling hierarchy. *Journal of Web Semantics*. 2015. №. 35. P. 214–234. DOI: 10.1016/j.websem.2015.05.002
4. jsld.org. jsld.org. URL: <https://jsld.org/> (дата звернення: 02.12.2024).
5. Kellogg G., Champin P.-A., Longley D. JSON-LD 1.1 – A JSON-based Serialization for Linked Data. 2020. URL: <https://hal.science/hal-02141614v2> (дата звернення: 02.12.2024).
6. ISO/IEC 24707:2018. ISO. URL: <https://www.iso.org/standard/66249.html> (дата звернення: 02.12.2024).
7. GitHub – gruninger/colore: Automatically exported from code.google.com/p/colore. GitHub. URL: <https://github.com/gruninger/colore> (дата звернення: 22.11.2024).
8. Hets (The heterogeneous tool set). Hets (The heterogeneous tool set). URL: <http://hets.eu/> (дата звернення: 22.11.2024).
9. GitHub – cmungall/cltools: tools for common-logic. GitHub. URL: <https://github.com/cmungall/cltools> (дата звернення: 22.11.2024).

10. Home - cyc. Cyc - The Next Generation of Enterprise AI. URL: <https://cyc.com/> (дата звернення: 22.11.2024).
11. Renssen V., Ashp. Gellish – A generic extensible ontological language – design and application of a universal data structure. Delft: Delft University Press. 2005. 238 p. URL: https://www.researchgate.net/publication/339529625_Gellish_A_Generic_Extensible_Ontological_Language_-_Design_and_Application_of_a_Universal_Data_Structure
12. Maniraj V., Sivakumar D. Ontology Languages – A Review. International Journal of Computer Theory and Engineering. 2010. №. 6, № 2. 887 p. URL: https://www.researchgate.net/publication/269801838_Ontology_Languages_-_A_Review (дата звернення: 22.11.2024).
13. Shen W. Multi-Agent systems for concurrent intelligent design and manufacturing. Taylor & Francis Group, 2019. 416 p.
14. OWL: A Large Language Model for IT Operations / H. Guo та ін. Computation and Language. 2023. URL: <https://doi.org/10.48550/arXiv.2309.09298> (дата звернення: 22.11.2024).
15. Koen de J. Transformation from OntoUML models to the OpenAPI Specification. 2024. URL: <https://purl.utwente.nl/essays/98294> (дата звернення: 22.11.2024).
16. Pareti P. A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs. *Reasoning Web. Declarative Artificial Intelligence*. 2021. P. 115–144. DOI: 10.1007/978-3-030-95481-9_6
17. Hoitash R., Hoitash U., Morris L. eXtensible Business Reporting Language (XBRL): A Review and Implications for Future Research. *AUDITING: A Journal of Practice & Theory*. 2021. Vol. 40, № 2. P. 107–132. DOI: 10.2308/ajpt-2019-517
18. A survey of RDF stores & SPARQL engines for querying knowledge graphs / W. Ali та ін. The VLDB Journal. 2021. № 31. P. 1–26. URL: <https://link.springer.com/article/10.1007/s00778-021-00711-3> (дата звернення: 22.11.2024).
19. Hozo – Ontology Editor. Hozo – Ontology Editor. URL: <https://www.hozo.jp/> (дата звернення: 22.11.2024).
20. GraphRAG for enterprise GenAI - Lettria. GraphRAG for enterprise GenAI – Lettria. URL: <https://www.lettria.com/> (дата звернення: 22.11.2024).
21. Protégé. protégé. URL: <https://protege.stanford.edu/> (дата звернення: 22.11.2024).
22. Onto4ALL – Ontology Graphical Editor. Onto4ALL – Ontology Graphical Editor. URL: <https://onto4all.com> (дата звернення: 22.11.2024).
23. FluentEditor – Ontology Editor Semantic Web. Cognitum Software House. URL: <https://www.cognitum.eu/semantics/fluenteditor/> (дата звернення: 22.11.2024).
24. Pertsas V., Constantopoulos P. Ontology-Driven Extraction of Contextualized Information from Research Publications. *15th International Conference on Knowledge Engineering and Ontology Development*, Rome, Italy, 13–15 листоп. 2023 p. DOI: 10.5220/0012254100003598
25. Salim M. N., Mustafa B. S. UTtoKB: a Model for Semantic Relation Extraction from Unstructured Text. *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey, 21–23 жовт. 2021 p. DOI: 10.1109/ismsit52890.2021.9604538
26. Demelo J., Sedig K. Forming Cognitive Maps of Ontologies Using Interactive Visualizations. *Multimodal Technologies and Interaction*. 2021. Vol. 5, № 1. 2 p. DOI: 10.3390/mti5010002 (дата звернення: 05.02.2025).
27. Dooley D., Nguyen M. H., Hsiao W. W. L. OntoTrek: 3D visualization of application ontology class hierarchies. *PLOS ONE*. 2023. Vol. 18, № 6. DOI: 10.1371/journal.pone.0286728
28. Brahmia Z., Grandi F., Bouaziz R. τSQL2: A TSQL2-Like Query Language for Temporal Ontologies Generated from JSON Big Data. *Big Data Mining and Analytics*. 2023. Vol. 6, № 3. P. 288–300. DOI: 10.26599/bdma.2022.9020044
29. Bolatito Y. A. An Enhanced OWL-Time Ontology for Complex Recurring Temporal Patterns / Y. A. Bolatito та ін. *Kasu Journal of Computer Science*. 2024. Vol. 1, № 2. P. 340–365. DOI: 10.47514/kjcs/2024.1.2.0013
30. Achich N. Approach to Reasoning about Uncertain Temporal Data in OWL 2, *Procedia Computer Science*. 2020. №. 176. P. 1141–1150. DOI:10.1016/j.procs.2020.09.110
31. Shanmurthy P. Augmentation of contextual knowledge based on domain dominant words for IoT applications interoperability. *Indonesian Journal of Electrical Engineering and Computer Science*. 2022. Vol. 27, № 1. 504 p. DOI: 10.11591/ijeecs.v27.i1.pp504-512
32. Ertuğrul D. Ç. A knowledge-based decision support system for inferring supportive treatment recommendations for diabetes mellitus. *Technology and Health Care*. 2023. P. 1–24. DOI: 10.3233/thc-230237
33. Saha R. Ontology-based intelligent decision support systems: A systematic approach/ *Web Semantics*. 2021. P. 177–193. DOI: 10.1016/b978-0-12-822468-7.00005-5
34. Spoladore D., Pessot E. Collaborative Ontology Engineering Methodologies for the Development of Decision Support Systems: Case Studies in the Healthcare Domain. *Electronics*. 2021. Vol. 10, № 9. 1060 p. DOI:10.3390/electronics10091060

35. Teixeira B. Application Ontology for Multi-Agent and Web-Services' Co-Simulation in Power and Energy Systems. *IEEE Access*. 2020. Vol. 8. P. 81129–81141. DOI: 10.1109/access.2020.2991010
36. Nachet B., Frendi M., Adla A. Physical Internet Enabled Traceability Systems for Sustainable Supply Chain Management. *Journal of information and organizational sciences*. 2024. Vol. 48, № 1. P. 99–116. DOI: 10.31341/jios.48.1.5
37. Aslam S., Vassilev V. T., Ouazzane K. Parallel Querying of Distributed Ontologies with Shared Vocabulary. *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*. 2019. Vol. 13, № 5. P. 287–294. DOI: [10.5281/zenodo.3298789](https://doi.org/10.5281/zenodo.3298789)
38. Query Optimization for Large Scale Clustered RDF Data / I. Zouaghi та ін. International Workshop on Data Warehousing and OLAP. 2020. P. 56–65. URL: <https://api.semanticscholar.org/CorpusID:212727545> (дата звернення: 05.02.2025).
39. Lin X., Jiang D. A Two-Phase Method for Optimization of the SPARQL Query. *Journal of Sensors*. 2022. Vol. 2022. P. 1–12. DOI: 10.1155/2022/4624856
40. Kang X. Grace: An Efficient Parallel SPARQL Query System over Large-Scale RDF Data. *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, м. Dalian, China, 5–7 трав. 2021 р. 2021. DOI: 10.1109/cscwd49262.2021.9437674
41. Agbaegbu J. Ontologies in Cloud Computing – Review and Future Directions. *Future Internet*. 2021. Vol. 13, № 12. 302 p. DOI: 10.3390/fi13120302
42. Karataiev O., Shubin I. Formal model of multi-agent architecture of a software system based on knowledge interpretation. *Radioelectronic and Computer Systems*. 2023. № 4. P. 53–64. DOI: 10.32620/reks.2023.4.05
43. Xie Y. Virtual Power Plants for Grid Resilience: A Concise Overview of Research and Applications. *IEEE/CAA Journal of Automatica Sinica*. 2024. Vol. 11, № 2. P. 329–343. DOI: 10.1109/jas.2024.124218
44. Ullah Z., Arshad A., Nekahi A. Virtual Power Plants: Challenges, Opportunities, and Profitability Assessment in Current Energy Markets. *Electricity*. 2024. Vol. 5, № 2. P. 370–384. DOI: 10.3390/electricity5020019
45. Zhang W. Virtual power plant integration with smart grids: a Review. *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, Nanjing, China, 27–29 трав. 2022 р. DOI: 10.1109/cieec54735.2022.9846220
46. Bassiliades N. A Tool for Transforming Semantic Web Rule Language to SPARQL Inferencing Notation. *International Journal on Semantic Web and Information Systems*. 2020. Vol. 16, № 1. P. 87–115. DOI: 10.4018/ijswis.2020010105
47. Calvanese D. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*. 2016. Vol. 8, № 3. P. 471–487. DOI.org/10.3233/sw-160217
48. Jajaga E., Ahmedi L. C-SWRL: SWRL for Reasoning over Stream Data. *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, San Diego, CA, USA, 30 січ. – 1 лют. 2017 р. DOI: 10.1109/icsc.2017.64
49. Dudar Z., Litvin S. Ontological description method for building service-oriented distributed learning systems. *Innovative technologies and scientific solutions for industries*. 2024. № 1 (27). P. 39–53. DOI: 10.30837/itssi.2024.27.039

References

1. Pouchard, L., Ivezic, N., Schlenoff, C. "Ontology engineering for distributed collaboration in manufacturing. AIS2000 Conf". 2000. 1012 p. available at: https://www.researchgate.net/publication/228596365_Ontology_engineering_for_distributed_collaboration_in_manufacturing (last accessed: 02.12.2024).
2. Guarino, N. "Ontologies and knowledge bases: towards a terminological clarification, Towards Very Large Knowl. Bases. Amsterdam:IOS Press". 1995. P. 25–32. available at: https://www.researchgate.net/publication/220041941_Ontologies_and_knowledge_bases_towards_a_terminological_clarification (last accessed: 02.12.2024).
3. Motz, R., Rohrer, E., Severi, P. (2015), "The description logicSHIQwith a flexible meta-modelling hierarchy". *Journal of Web Semantics*. 2015. Vol. 35. P. 214–234. DOI: 10.1016/j.websem.2015.05.002
4. "jsld.org. jsld.org". available at: <https://jsld.org/> (last accessed: 02.12.2024).
5. Kellogg, G., Champin, P.A., Longley, D. "JSON-LD 1.1 – A JSON-based Serialization for Linked Data". 2020. available at: <https://hal.science/hal-02141614v2> (last accessed: 02.12.2024).
6. "ISO/IEC 24707:2018. ISO". available at: <https://www.iso.org/standard/66249.html> (last accessed: 02.12.2024).
7. "GitHub – gruninger/colore: Automatically exported from code.google.com/p/colore. GitHub". available at: <https://github.com/gruninger/colore> (last accessed: 22.11.2024).
8. "Hets (The heterogeneous tool set). Hets (The heterogeneous tool set)". available at: <http://hets.eu/> (last accessed: 22.11.2024).

9. "GitHub – cmungall/cltools: tools for common-logic. GitHub". available at: <https://github.com/cmungall/cltools> (last accessed: 22.11.2024).
10. "Home – cyc. Cyc – The Next Generation of Enterprise AI". available at: <https://cyc.com/> (last accessed: 22.11.2024).
11. Renssen, V., "Ashp. Gellish – A generic extensible ontological language – design and application of a universal data structure. Delft: Delft University Press". 2005. 238 p. available at: https://www.researchgate.net/publication/339529625_Gellish_A_Generic_Extensible_Ontological_Language_-_Design_and_Application_of_a_Universal_Data_Structure
12. Maniraj, V., Sivakumar, D. "Ontology Languages – A Review. International Journal of Computer Theory and Engineering". 2010. Vol. 6, no. 2. P. 887. available at: https://www.researchgate.net/publication/269801838_Ontology_Languages_-_A_Review (last accessed: 22.11.2024).
13. Shen, W. "Multi-Agent systems for concurrent intelligent design and manufacturing. Taylor & Francis Group", 2019. 416 p.
14. "OWL: A Large Language Model for IT Operations / H. Guo et al. Computation and Language". 2023. available at: <https://doi.org/10.48550/arXiv.2309.09298> (last accessed: 22.11.2024).
15. Koen de, J. "Transformation from OntoUML models to the OpenAPI Specification". 2024. available at: <https://purl.utwente.nl/essays/98294> (last accessed: 22.11.2024).
16. Pareti, P. (2021), "A Review of SHACL: From Data Validation to Schema Reasoning for RDF Graphs". *Reasoning Web. Declarative Artificial Intelligence*. 2021. P. 115–144. DOI: 10.1007/978-3-030-95481-9_6
17. Hoitash, R., Hoitash, U., Morris, L. (2021), "eXtensible Business Reporting Language (XBRL): A Review and Implications for Future Research". *AUDITING: A Journal of Practice & Theory*. 2021. Vol. 40, No. 2. P. 107–132. DOI: 10.2308/ajpt-2019-517
18. "A survey of RDF stores & SPARQL engines for querying knowledge graphs / W. Ali et al. The VLDB Journal". 2021. No. 31. P. 1–26. available at: <https://link.springer.com/article/10.1007/s00778-021-00711-3> (last accessed: 22.11.2024)
19. "Hozo – Ontology Editor. Hozo – Ontology Editor". available at: <https://www.hozo.jp/> (last accessed: 22.11.2024).
20. "GraphRAG for enterprise GenAI – Lettria. GraphRAG for enterprise GenAI - Lettria". available at: <https://www.lettria.com/> (last accessed: 22.11.2024).
21. "Protégé. protégé". available at: <https://protege.stanford.edu/> (last accessed: 22.11.2024).
22. "Onto4ALL – Ontology Graphical Editor. Onto4ALL – Ontology Graphical Editor". available at: <https://onto4all.com> (last accessed: 22.11.2024).
23. "FluentEditor – Ontology Editor Semantic Web. Cognitum Software House". available at: <https://www.cognitum.eu/semantics/fluenteditor/> (last accessed: 22.11.2024).
24. Pertsas, V., Constantopoulos, P. (2023), "Ontology-Driven Extraction of Contextualized Information from Research Publications". *15th International Conference on Knowledge Engineering and Ontology Development*, Rome, Italy, 13–15 November 2023. DOI: 10.5220/0012254100003598
25. Salim, M. N., Mustafa, B. S. (2021), "UTtoKB: a Model for Semantic Relation Extraction from Unstructured Text". 2021 *5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey, 21–23 October 2021. DOI: 10.1109/ismsit52890.2021.9604538
26. Demelo, J., Sedig, K. (2021), "Forming Cognitive Maps of Ontologies Using Interactive Visualizations". *Multimodal Technologies and Interaction*. 2021. Vol. 5, No. 1. 2 p. DOI: 10.3390/mti5010002
27. Dooley, D., Nguyen, M. H., Hsiao, W. W. L. (2023), "OntoTrek: 3D visualization of application ontology class hierarchies". *PLOS ONE*. 2023. Vol. 18, No. 6. DOI: 10.1371/journal.pone.0286728
28. Brahmia, Z., Grandi, F., Bouaziz, R. (2023), "τSQL: A TSQL2-Like Query Language for Temporal Ontologies Generated from JSON Big Data". *Big Data Mining and Analytics*. 2023. Vol. 6, No. 3. P. 288–300. DOI: 10.26599/bdma.2022.9020044
29. Bolatito, Y. A. (2024), "An Enhanced OWL-Time Ontology for Complex Recurring Temporal Patterns" / et al. *Kasu Journal of Computer Science*. 2024. Vol. 1, No. 2. P. 340–365. DOI: 10.47514/kjcs/2024.1.2.0013
30. Achich, N. (2020), "Approach to Reasoning about Uncertain Temporal Data in OWL 2", *Procedia Computer Science*. 2020. Vol. 176. P. 1141–1150. DOI:10.1016/j.procs.2020.09.110
31. Shanmurthy, P. (2022), "Augmentation of contextual knowledge based on domain dominant words for IoT applications interoperability", *Indonesian Journal of Electrical Engineering and Computer Science*. 2022. Vol. 27, No. 1. 504 p. DOI: 10.11591/ijeecs.v27.i1.pp504-512
32. Ertuğrul, D. Ç. (2023), "A knowledge-based decision support system for inferring supportive treatment recommendations for diabetes mellitus", *Technology and Health Care*. 2023. P. 1–24. DOI: 10.3233/thc-230237
33. Saha, R. (2021), "Ontology-based intelligent decision support systems: A systematic approach". *Web Semantics*. 2021. P. 177–193. DOI: 10.1016/b978-0-12-822468-7.00005-5

34. Spoladore, D., Pessot, E. (2021), "Collaborative Ontology Engineering Methodologies for the Development of Decision Support Systems: Case Studies in the Healthcare Domain". *Electronics*. 2021. Vol. 10, No. 9. 1060 p. DOI:10.3390/electronics10091060
35. Teixeira, B. (2020), "Application Ontology for Multi-Agent and Web-Services' Co-Simulation in Power and Energy Systems", *IEEE Access*. 2020. Vol. 8. P. 81129–81141. DOI: 10.1109/access.2020.2991010
36. Nachet, B., Frendi, M., Adla, A. (2024), "Physical Internet Enabled Traceability Systems for Sustainable Supply Chain Management". *Journal of information and organizational sciences*. 2024. Vol. 48, No. 1. P. 99–116. DOI: 10.31341/jios.48.1.5
37. Aslam, S., Vassilev, V. T., Ouazzane, K. (2019), "Parallel Querying of Distributed Ontologies with Shared Vocabulary". *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*. 2019. Vol. 13, No. 5. P. 287–294. DOI: 10.5281/zenodo.3298789
38. "Query Optimization for Large Scale Clustered RDF Data / I. Zouaghi et al". *International Workshop on Data Warehousing and OLAP*. 2020. P. 56–65. available at: <https://api.semanticscholar.org/CorpusID:212727545> (last accessed: 05.02.2025).
39. Lin, X., Jiang, D. (2022), "A Two-Phase Method for Optimization of the SPARQL Query". *Journal of Sensors*. 2022. Vol. 2022. P. 1–12. DOI: 10.1155/2022/4624856
40. Kang, X. (2021), "Grace: An Efficient Parallel SPARQL Query System over Large-Scale RDF Data", *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Dalian, China, 5 – 7 May 2021. DOI: 10.1109/cscwd49262.2021.9437674
41. Agbaegbu, J. (2021), "Ontologies in Cloud Computing – Review and Future Directions", *Future Internet*. 2021. Vol. 13, No. 12. 302 p. DOI: 10.3390/fi13120302
42. Karataiev, O., Shubin, I. (2023), "Formal model of multi-agent architecture of a software system based on knowledge interpretation". *Radioelectronic and Computer Systems*. 2023. No. 4. P. 53–64. DOI: 10.32620/reks.2023.4.05
43. Xie, Y. (2024), "Virtual Power Plants for Grid Resilience: A Concise Overview of Research and Applications". *IEEE/CAA Journal of Automatica Sinica*. 2024. Vol. 11, No. 2. P. 329–343. DOI: 10.1109/jas.2024.124218
44. Ullah, Z., Arshad, A., Nekahi, A. (2024), "Virtual Power Plants: Challenges, Opportunities, and Profitability Assessment in Current Energy Markets". *Electricity*. 2024. Vol. 5, No. 2. P. 370–384. DOI: 10.3390/electricity5020019
45. Zhang, W. (2022), "Virtual power plant integration with smart grids: a Review", *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, Nanjing, China, 27–29 May 2022. DOI: 10.1109/cieec54735.2022.9846220
46. Bassiliades, N. (2020), "A Tool for Transforming Semantic Web Rule Language to SPARQL Inferencing Notation". *International Journal on Semantic Web and Information Systems*. 2020. Vol. 16, No. 1. P. 87–115. DOI: 10.4018/ijswis.2020010105
47. Calvanese, D. (2016), "Ontop: Answering SPARQL queries over relational databases", *Semantic Web*. 2016. Vol. 8, No. 3. P. 471–487. DOI.org/10.3233/sw-160217
48. Jajaga, E., Ahmedi, L. (2017), "C-SWRL: SWRL for Reasoning over Stream Data". *2017 IEEE 11th International Conference on Semantic Computing (ICSC)*, San Diego, CA, USA, 30 January – 1 February 2017. DOI: 10.1109/icsc.2017.64
49. Dudar, Z., Litvin, S. (2024), "Ontological description method for building service-oriented distributed learning systems". *Innovative technologies and scientific solutions for industries*. 2024. No. 1 (27). P. 39–53. DOI: 10.30837/itssi.2024.27.039

Надійшла (Received) 10.01.2025

Відомості про авторів / About the Authors

Долгий Андрій Іванович – Харківський національний університет радіоелектроніки, магістрант кафедри програмної інженерії, Харків, Україна; e-mail: andrii.dolhyi@nure.ua, ORCID ID: <https://orcid.org/0009-0005-5412-891X>

Шубін Ігор Юрійович – кандидат технічних наук, доцент, Харківський національний університет радіоелектроніки, професор кафедри програмної інженерії, заступник начальника навчально-методичного відділу, Харків, Україна; e-mail: igor.shubin@nure.ua, ORCID ID: <https://orcid.org/0000-0002-1073-023X>

Dolhyi Andrii – Kharkiv National University of Radio Electronics, Master's Student at the Department of Software Engineering, Kharkiv, Ukraine.

Shubin Ihor – PhD (Engineering Sciences), Associate Professor, Kharkiv National University of Radio Electronics, Professor at the Department of Software Engineering, Deputy Head at the Educational and Methodological Department, Kharkiv, Ukraine.

RESEARCH OF MODERN METHODS AND TOOLS OF ONTOLOGICAL ENGINEERING IN THE CONTEXT OF CREATING INTELLECTUAL SYSTEMS

The subject of the work is the study of modern methods and tools of ontological engineering in the context of developing intelligent systems. In particular, ontological engineering as a process of creating formalized knowledge bases for intelligent systems. The work analyzes ontological modeling languages (RDF, OWL, SWRL, etc.), the SPARQL query language and tools such as Protégé, Hozo, etc., which allow implementing the specified approach. **The purpose** of the research is to identify and analyze modern tools and methods of ontological engineering for developing knowledge bases that provide high prediction accuracy and adaptability in complex intelligent systems using artificial intelligence or other prediction algorithms. **The objective** of this study is to conduct a comprehensive analysis of modern methods and tools of ontology engineering, including a comparative description of their capabilities and limitations when applied in intelligent systems. The study involves studying methods for automating the processes of creating and updating ontologies, in particular using deep learning and natural language processing, as well as evaluating promising languages and tools for modeling ontology. Particular attention is paid to the analysis of the application of ontologies in various subject areas where high forecasting accuracy is required, with further research into methods for optimizing queries to ontological databases. The practical part of the work involves creating a demonstration ontological knowledge base for an intelligent control system for virtual distributed power plants, which will allow assessing the practical value of the study. The research **methods** include a thorough analysis of the literature and available documentation on the topic, a comparative analysis of the obtained data and further demonstration in practice of the application of selected tools and methods, based on their indicators. **Results** The issue of the feasibility of ontological engineering in modern intelligent systems is considered. Tools and methods for creating knowledge bases based on ontologies are presented and analyzed. The paper provides comparative statistics of modern tools for creating ontologies. A substantiated vision of the situation around the tools and methods of ontological engineering was put forward. The limitations and unique aspects of each of the considered tools are revealed. Optimal approaches to creating knowledge bases for intelligent systems are determined. The use and features of ontological engineering tools are demonstrated on a practical example in the context of intelligent systems for controlling virtual distributed power plants. **In conclusion**, the novelty of the study lies in the modernity of the view on the issue of choosing tools for creating knowledge bases using ontological engineering, in particular in terms of their use in intelligent systems, in combination with artificial intelligence. The results of this work can be used in the development of relevant intelligent systems that use knowledge bases. The study provides a wide range of information and comparisons of ontological engineering approaches and tools.

Keywords: ontology engineering; semantic modeling; knowledge bases; ontology languages; ontology tools.

Бібліографічні описи / Bibliographic descriptions

Долгий А. И., Шубин И. Ю. Дослідження сучасних методів та інструментів онтологічного інжинірингу в контексті створення інтелектуальних систем. *Сучасний стан наукових досліджень та технологій в промисловості*. 2025. № 1 (31). С. 32–48. DOI: <https://doi.org/10.30837/2522-9818.2025.1.032>

Dolhyi, A., Shubin, I. (2025), "Research of modern methods and tools of ontological engineering in the context of creating intellectual systems", *Innovative Technologies and Scientific Solutions for Industries*, No. 1 (31), P. 32–48. DOI: <https://doi.org/10.30837/2522-9818.2025.1.032>
