УДК 004.896 https://doi.org/10.15407/pp2020.02-03.050

МЕТОДИ ТА ІНСТРУМЕНТИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВАЛІДАЦІЇ РЕЗУЛЬТАТІВ НЕФОРМАЛЬНОГО НАВЧАННЯ

С.М. Прийма, О.В. Строкань, Ю.В. Рогушина, А.Я. Гладун, А.А. Мозговенко

У публікації розглянуті методи та інструменти розробки інформаційної системи на основі комп'ютерних онтологій. Прототипом комп'ютерної онтології виступає онтологія багатомовного класифікатора навичок ESCO. Представлені етапи розробки інформаційної системи валідації результатів неформального навчання. Проаналізовано сучасні методи проектування та розробки застосувань, призначених для функціонування у відкритому середовищі Web. Автори пропонують інноваційний метод створення інформаційної системи, який базується на використанні основних елементів проекту Semantic Web. В роботі наведено опис інструментів, які використані при розробці інформаційної системи валідації результатів неформального навчання, зокрема проаналізована система управління базами даних Neo4j, робота конекторів та запитів SPARQL до даних, розміщених в RDF-сховищі, охарактеризовані засоби створення веб-серверу, виконаний порівняльний аналіз PHP фреймворки Laravel і Symfony для розробки веб-додатку.

Ключові слова: валідація, інформаційна система, неформальне навчання, GraphDB, онтологія, Semantic Web, мова запитів SPARQL, конектор, RDF-сховище.

В публикации рассмотрены методы и инструменты разработки информационной системы на основе компьютерных онтологий. Прототипом компьютерной онтологии выступает онтология многоязычного классификатора навыков ESCO. Представлены этапы разработки информационной системы валидации результатов неформального обучения. Проанализированы современные методы проектирования и разработки приложений, предназначенных для функционирования в открытой среде Web. Авторы предлагают инновационный метод создания информационной системы, основанный на использовании основных элементов проекта Semantic Web. В работе приведено описание инструментов, которые использованы при разработке информационной системы валидации результатов неформального обучения, в частности проанализирована система управления базами данных Neo4j, работа коннекторов и запросов SPARQL к данным, размещенным в RDF-хранилище, охарактеризованы средства создания веб-сервера, выполненный сравнительный анализ PHP фреймворков Laravel и Symfony для разработки веб-приложения.

Ключевые слова: валидация, информационная система, неформальное обучение, GraphDB, онтология, Semantic Web, язык запросов SPARQL, коннектор, RDF-хранилище.

The publication describes methods and tools for development an information system (IS) for acceptance of the non-formal and informal learning results on base of computer ontologies. ESCO ontology of the multilingual skills classifier is used as a prototype of domain ontology that provides knowledge for developed IS. We propose the main stages of development of IS for validation of the results of non-formal learning that include creation of an ontology schema, process of integration of the obtained ontology into the RDF repository, development of application architecture and creation of user interface.

The modern approaches to design and development of knowledge-oriented distributed applications intended for functioning in the open Web environment are analyzed, the existing methods and software tools for the presentation and analysis of knowledge and their suitability for solving the task are considered. We propose an innovative method of IS development based on the use of such elements of the Semantic Web project as ontologies, Web services and software agents.

The paper describes the tools used in process of development of IS for validation of non-formal learning outcomes. In particular, we analyse Neo4j database management system serving the GraphDB database, the specifics of connectors and SPARQL requests to the data stored in the RDF repository and the tools used for web server creation. Comparison of PHP frameworks for web applications is performed in consideration of task requirements.

Functional modeling of IS in order to determine its main functionalities is performed, and the DFD data flow diagrams of system are designed. The benefits of Laravel software are established on base of the analysis of such criteria as security, readiness to installation of plugins and libraries, support for the MVC (Model-View-Controller) concept. User interface is developed to ensure user dialogue with IS. Authers analyse software tools oriented on development of user interface and select React framework that works efficiently with all software tools selected for IS development on the previous stages of analysis.

Key words: validation, information system, non-formal learning, GraphDB, ontology, Semantic Web, SPARQL query language, connector, RDF storage.

Вступ

У зв’язку зі стрімким розвитком систем, заснованих на знаннях, які займають широку область в інформаційних технологіях, формуючи власні методи і принципи, значного розвитку зазнала ініціатива Semantic Web [1, 2]. Основними особливостями технології Semantic Web [3] є розширення машинної обробки інформації, а саме: інтелектуальне оброблення, засноване на семантичних технологіях. У найбільш узагальненому розумінні семантична ідентифікація певного фрагменту даних полягає у встановленні його зв’язку з елементом опису знань предметної області та явним визначенням змісту такого зв’язку. Сьогодні для цього широко застосовуються онтології [4]. Наприклад, онтологія певної предметної області може бути використана для представлення результатів навчання в цій області, якщо окремі фрагменти резюме потенційного працівника пов’язані із поняттями такої онтології тими відношеннями, що також формалізовані засобами саме цієї онтології.

У відповідності із вищезазначеним постає необхідність у розробці інформаційної системи валідації результатів неформального навчання, яка б легко адаптувалася до предметної області і забезпечувала оптимізацію процесу поєднання ринку освітніх послуг і ринку праці.

Використання технологій Semantic Web для семантичної ідентифікації результатів неформального навчання

Одним з найбільш популярних проектів, пов'язаних із обробкою розподілених знань, є Semantic Web. Концепцію Semantic Web висунув Тім Бернерс-Лі − один з основоположників World Wide Web і голова консорціуму (W3C) в роботі «Semantic Web» [5]. Метою цього проекту є перетворення усієї сукупності наявних інформаційних ресурсів, досяжних через Web, на розподілену гетерогенну базу знань. Загальна ідея концепції Semantic Web полягає в організації такого представлення даних у мережі, щоб допускалася не тільки візуалізація, але й ефективна автоматична обробка даних програмами різних виробників. Основними компонентами Semantic Web є онтології, Web-сервіси та програмні агенти. Для їх подання в рамках Semantic Web розроблені наступні відкриті стандарти подання знань, як мова подання онтологій OWL, стандарт опису метаданих IP RDF, та мова запитів до цих формалізованих знань SPARQL.

На сьогодні проект Semantic Web активно розвивається, з'являються нові мови, стандарти та інструментальні засоби, а також удосконалюються наявні. Тому доцільно в процесі розробки інтелектуальної прикладної системи, що базується на використанні ресурсів Web, орієнтуватися саме на ці результати і створювати семантичні Web-сервіси, що можуть ефективно використовувати всі переваги нового інформаційного середовища. Семантичні Web-сервіси розширюють поняття звичайних Web-сервісів у частині використання семантичної інформації, а саме онтологій та семантичної розмітки як для прикладних, так і для системних потреб.

Використання онтологічного аналізу забезпечує можливість перенесення знань до нових застосувань, автоматизованого експорту відомостей із семантично розмічених інформаційних ресурсів та побудови спільної термінологічної основи для взаємодії між різними ресурсами та інформаційними системами. Технології Semantic Web легко інтегруються з іншими сучасними Web-технологіями, такими як інтелектуальні Wiki (приклад – Semantic MediaWiki) [4, 6].

Зважаючи на це, пропонуємо інформаційну систему валідації результатів неформального навчання, призначення якої полягає у формуванні паспорту набутих компетенцій, пошуку вакансій та співставлення компетенцій з вимогами до вакансії на основі моделі ESCO [7].

При розробці системи враховані наступні технології Semantic Web: наявність персональних програмних агентів для кожного з потенційних роботодавців та пошукачів вакансій для персоніфікації взаємодії; підтримка функцій реєстрації та пошуку вакансій, реєстрація та пошук резюме, співставлення вакансій та резюме на семантичному рівні, пошук освітніх сервісів, спроможних надати певну кваліфікацію або освіту, співставлення навчальних курсів та програм із професіями; забезпечення взаємозв'язку між професіями, знаннями, навичками, компетенціями та кваліфікаціями, а також між їх характеристиками формалізуються за допомогою онтології.

Засоби створення інформаційної системи валідації результатів неформального навчання

Для визначення основних функціональних можливостей інформаційної системи виконаємо функціональне моделювання IDEF0. IDEF0 − методологія функціонального моделювання і графічна нотація, призначена для формалізації і опису бізнес-процесів. Відмінною особливістю IDEF0 є її акцент на підпорядкованість об'єктів. У IDEF0 розглядаються логічні відносини між роботами, а не їх тимчасова послідовність (потік робіт).

На рис. 1 показана функціональна модель IDEF0 інформаційної системи валідації результатів неформального навчання.

Функціональна модель IDEF0 розробленої інформаційної системи складається з таких основних компонентів:

- головний блок − інформаційна система на основі комп'ютерних онтологій. Прототипом комп'ютерної онтології виступає онтологія багатомовного класифікатора навичок ESCO;
- вхідними даними є «Вхідні дані від ESCO» − онтологія ESCO;
- елементами управління, які безпосередньо впливають на розробку архітектури та технічного завдання інформаційної системи, є: «Нормативи, Положення, Акти, Розпорядження, ГОСТи, ДСТУ» та «Керівник проекту»;
- в ролі механізмів при розробці системи виступають: пошукувач, роботодавець, центр зайнятості, експерт та провайдер освітніх послуг;
- вихідними даними системи є таргетовані дані для пошукувача та таргетовані дані для роботодавця.

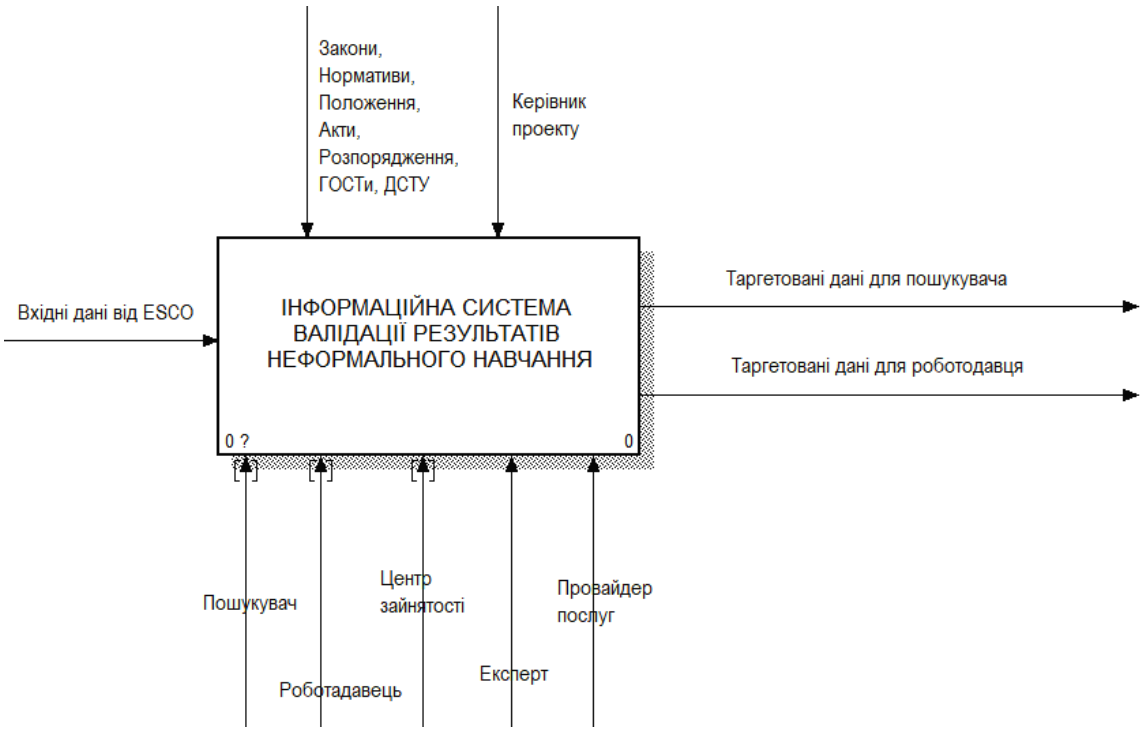

Рис. 1. Функціональна модель IDEF0 системи

Після розробки функціональної модель IDEF0 переходимо до наступного етапу, а саме до проектування діаграми потоків даних системи DFD. Для цього проведемо декомпозицію функціональної моделі IDEF0. Діаграма потоків даних DFD являє собою модель проектування, графічне представлення «потоків» даних в інформаційній системі. По суті так називається методологія графічного структурного аналізу, яка описує зовнішні по відношенню до системи джерела та адресати даних, логічні функції, потоки даних та сховища даних, до яких здійснюється доступ. Діаграми потоків даних містять чотири типи графічних елементів: процеси – являють собою трансформацію даних в рамках описуваної системи; сховища даних (репозиторії); зовнішні по відношенню до системи сутності; потоки даних між елементами системи.

Декомпозиція функціональної моделі інформаційної системи (рис. 1) показана на рис. 2.

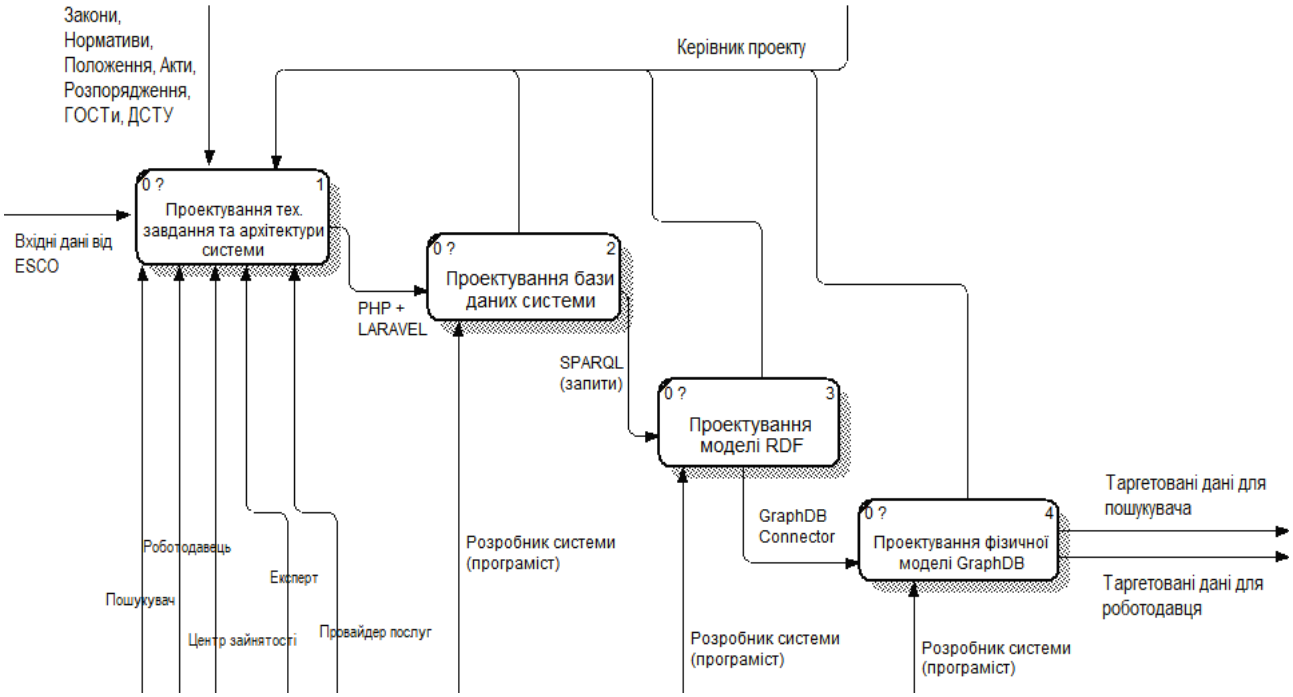

Рис. 2. Декомпозиція функціональної моделі системи

Декомпозиція функціональної моделі розроблюваної системи містить чотири блоки: проектування технічного завдання та архітектури системи; проектування бази даних системи; проектування моделі RDF; проектування фізичної моделі GraphDB.

Реалізація заявлених функцій забезпечується використанням невеликого набору універсальних засобів, за допомогою яких можна оперувати будь-якою інформацією предметної області: перегляд інформації про екземпляри (професії, навички, компетенції тощо), пошук екземплярів з необхідними властивостями та редагування екземплярів.

В наш час для опису онтологій та екземплярів понять в концепції семантичного павутиння використовується мова OWL (Web Ontology Language) – синтаксичний варіант дескрипційної логіки [7, 8, 9, 10]. Дана мова дозволяє описувати класи і відношення між ними.

Серед нових методів обробки даних виділяється модель RDF (Resource Description Framework) − формалізм опису взаємопов'язаних сутностей. RDF-сховища використовуються при вирішені задач, в яких дані можуть мати непередбачувану кількість зв'язків. Вони базуються на стандартах комітету W3C для мови опису графів та для обробки графових даних (SPARQL). RDF визначає загальну архітектуру метаданих і призначене для забезпечення сумісності метаданих за допомогою спільної семантики, структури і синтаксису. Основою RDF є подання даних у вигляді тверджень (трійок, триплет) суб'єкт-предикат-об'єкт, що описують спрямовану зв'язок від суб'єкта до об'єкта. Для ідентифікації суб'єктів, об'єктів і предикатів використовується ідентифікатор Uniform Resource Identifier (URI) [12], що є узагальненням поняття URL. Один і той же URI може в різних триплетах перебувати в різних позиціях: бути і суб'єктом, і предикатом, і об'єктом; тим самим триплети утворюють свого роду граф, званий RDF-графом.

Для роботи з RDF-даними існує мова запитів SPARQL. SPARQL − мова запитів до даних, представлених по моделі RDF, а також протокол для передачі цих запитів і відповідей на них. SPARQL є рекомендацією консорціуму W3C і однією з технологій семантичного павутиння. Представлення SPARQL-точок доступу (SPARQL endpoint) є рекомендованою практикою при публікації даних у всесвітньому павутинні. SPARQL володіє можливостями формувати запити до обов'язкових і необов'язкових графових шаблонів разом із кон'юнкіями і диз'юнкціями. Результат запитів SPARQL може бути представлений у вигляді результуючих наборів або RDF-графів [8].

Більшість запитів SPARQL включає набір шаблонів триплетів, який називається основним графовим шаблоном. Шаблони триплетів подібні RDF-триплетам, за виключенням того, що кожний суб'єкт, предикати об'єкт може бути змінною. Основний графовий шаблон відповідає підграфу RDF-даних, коли RDF-терміни цього підграфу можуть бути замінені змінними, а результат є RDF-графом, еквівалентним підграфу.

Відповідно до [13] розроблення інформаційної системи валідації результатів неформального навчання, яка передбачає застосування онтологічних баз, виконується у чотири етапи: створення схеми онтології; інтеграція онтології в RDF-сховище; розробка архітектури додатку; розробка інтерфейсу інформаційної системи.

Перший етап розробки системи полягає у створенні онтології, як засобу представлення знань. Онтологія утворює систему, що складається з наборів понять і тверджень, на основі яких можна будувати класи, об'єкти і відношення. Вона визначає семантику предметної області й сприяє встановленню зв'язків між значеннями її елементів.

Для роботи системи використана онтологія європейського класифікатора ESCO. Для модифікації вихідної онтології предметної області та її зберігання використана системи зберігання метаданих Protege. Платформа Protege підтримує два основних способи моделювання онтологій за допомогою редакторів Protege - Frames і Protege-OWL. Для візуалізації онтології використана система управління графовими базами даних Ne04j.

На рис.3 показано укрупнений граф онтології, завантаженої з ESCO, з описом головних зв'язків. На укрупненому графі виділені чотири класи: «Skill», «Occupation», «NodeLiteral» та «Label», так як вони беруть безпосередню участь при роботі з онтологією, інші є або описовими або концептуальними.

Клас «Skill» представляє в онтології навички, клас «Occupation» − професії, клас «NodeLiteral» − опис професій та навичок, а клас «Label» − назви професій та навичок.

Другий етап полягає в інтеграції онтології в RDF-сховище. Для зберігання отриманої на першому етапі онтології скористуємося базою даних семантичних графів GraphDB. GraphDB − це база даних графів, сумісна зі специфікаціями RDF та SPARQL. Вона підтримує відкриті API на основі проекту RDF4J, дозволяє швидко публікувати пов'язані дані в Інтернеті і використовується для зв'язування даних з різних джерел, подальшої їх індексації для семантичного пошуку та збагачує цю інформацію за допомогою аналізу текста для побудови великих згарів знань.

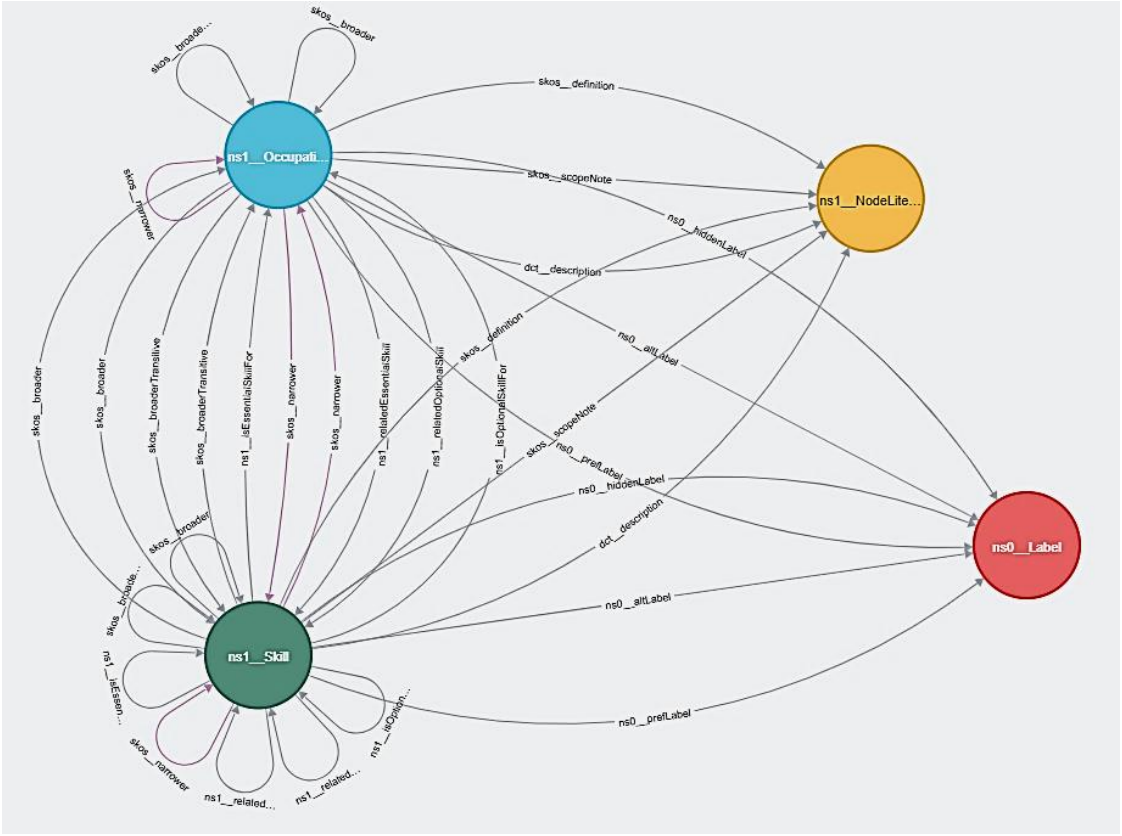

Рис. 3. Укрупнений граф отриманий в редакторі онтологій Protege

Для роботи з RDF-сховищем необхідно звертатися до нього через мову запитів SPARQL. На рис. 4 показано приклад SPARQL-запиту, який дозволяє визначити список навичок.

Запит складається з двох частин: умова SELECT задає змінні, які повинні відображатися в результатах запиту, а умова WHERE надає основний графовий шаблон, якому повинні відповідати графові дані. У даному прикладі основний графовий шаблон складається з одного шаблону триплету зі змінними ?skill ?o ?p на місці об'єкта.

```
PREFIX skos-xl: <http://www.w3.org/2008/05/skos-xl#>
PREFIX skos:  <http://www.w3.org/2004/02/skos/core#>
SELECT DISTINCT ?skill ?o ?p #?label
WHERE {
  ?skill a esco:Skill.
  ?skill skos:prefLabel ?o.
  ?skill skos:altLabel ?p.
    FILTER (lang(?o)='en' )
    FILTER (lang(?p)='en' )
```

Рис. 4. Приклад SPARQL-запиту

В результаті роботи запиту буде отримано список навичок з їх написами на англійській мові (рис. 5).

11	http://data.europa.eu/esco/skill/a01243ed-cd31-44d5-b8ee-b19f6718425e	"manage cage nets"@en	"managing cage nets"@en
12	http://data.europa.eu/esco/skill/a01cf14b-f912-4b5b-a1a4-b4a2182d6f21	"network within the writing industry"@en	"collaborate within writing industry"@en
13	http://data.europa.eu/esco/skill/a01cf14b-f912-4b5b-a1a4-b4a2182d6f21	"network within the writing industry"@en	"networking within writing industry"@en
14	http://data.europa.eu/esco/skill/a01cf14b-f912-4b5b-a1a4-b4a2182d6f21	"network within the writing industry"@en	"teaching writing"@en
15	http://data.europa.eu/esco/skill/a01cf14b-f912-4b5b-a1a4-b4a2182d6f21	"network within the writing industry"@en	"work together within writing industry"@en
16	http://data.europa.eu/esco/skill/a0225b42-4c3a-4551-962b-9facffa07d4c	"leasing process"@en	"rental arrangements"@en
17	http://data.europa.eu/esco/skill/a0225b42-4c3a-4551-962b-9facffa07d4c	"leasing process"@en	"leasing of property"@en
18	http://data.europa.eu/esco/skill/a0225b42-4c3a-4551-962b-9facffa07d4c	"leasing process"@en	"rental agreements"@en

Рис. 5. Результат роботи SPARQL-запиту

Для прискорення роботи зі сховищем доцільно використовувати конектори. Конектори GraphDB забезпечують швидкий звичайний і гранований пошук (агрегація), зазвичай реалізується зовнішнім компонентом або службою, такою як Lucene.

На рис. 6 показано приклад створення конектора в GraphDB. Задаємо йому ім'я і поля.

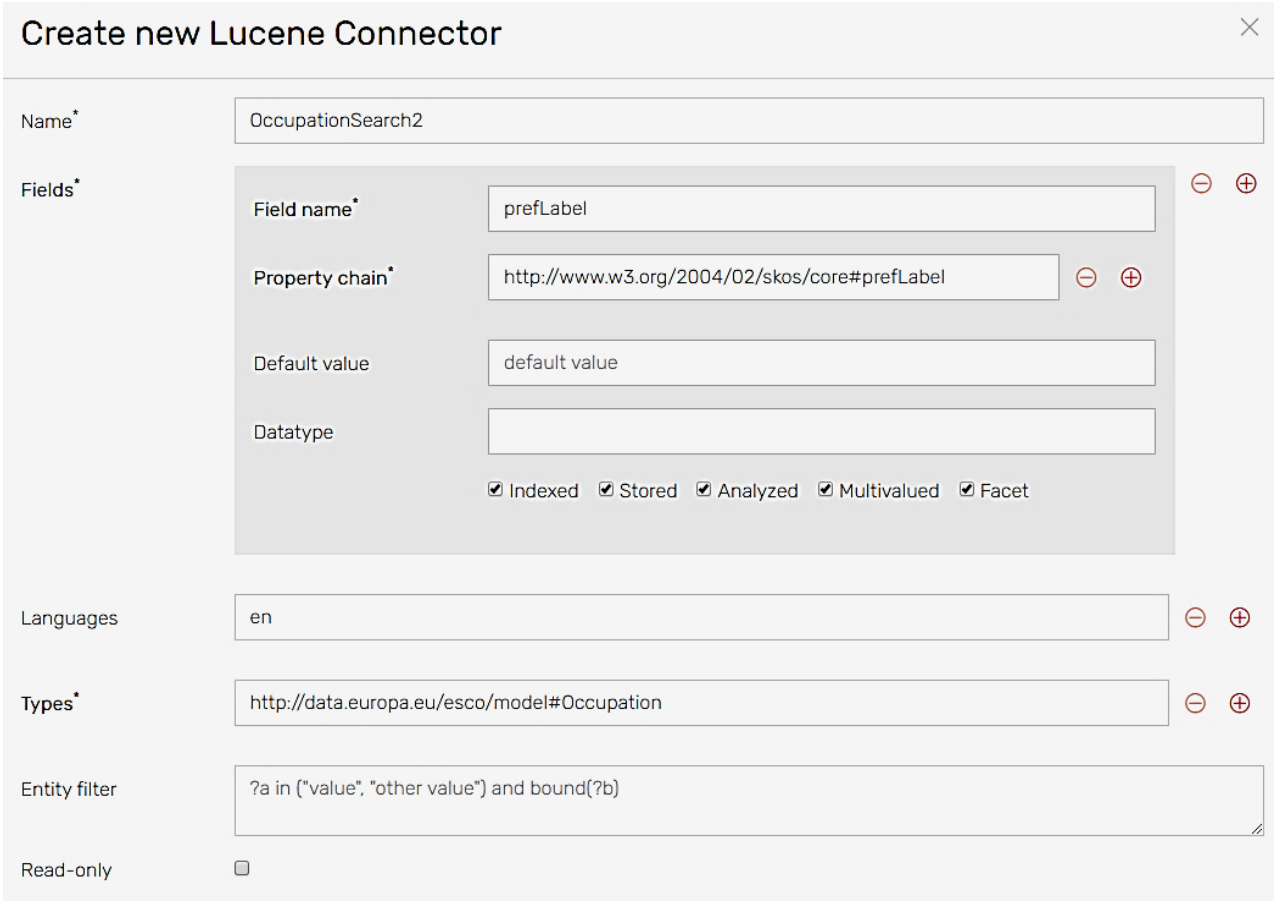

Рис. 6. Форма для створення конектора в GraphDB

Конектори GraphDB мають додаткову перевагу автоматичного оновлення даних із сховищами GraphDB. Конектори забезпечують синхронізацію на рівні сутності, де сутність визначається як така, що має унікальний ідентифікатор (URI) та набір властивостей і значень властивостей. З точки зору RDF, це відповідає набору трійці, що мають один і той же предмет. Крім простих властивостей (визначених однією трійкою), конектори підтримують ланцюги властивостей. Ланцюг властивостей − це послідовність трійки, де об'єкт кожної трійки є предметом наступної трійки.

Даний конектор приймає текст у полі і виконує пошук за визначеними мітками і повертає посилання на клас, який відповідає заданому запиту.

Для перевірки роботи створеного конектора виконуємо запит (рис. 7) і отримуємо результат роботи запиту (рис. 8).

```
1  PREFIX :<http://www.ontotext.com/connectors/lucene#>
2  PREFIX inst:<http://www.ontotext.com/connectors/lucene/instance#>
3
4  SELECT ?entity {
5    ?search a inst:OccupationSearch2 ;
6        :query "terrazzo setter" ;
7        :entities ?entity .
8  }
```

Рис. 7. Перевірка конектора в GraphDB

	entity
1	http://data.europa.eu/esco/occupation/a50d64e7-2dba-4418-81a7-92b2ba2e508f
2	http://data.europa.eu/esco/occupation/8249697a-3dc8-417d-9acf-3a7e0864069a
3	http://data.europa.eu/esco/occupation/6729aa5a-285f-4baf-a466-c2ccd2f2d96f
4	http://data.europa.eu/esco/occupation/02447817-ea01-4d8b-b09c-8bc128e447e6

Рис. 8. Результат роботи конектора

Конектори GraphDB володіють такими властивостями [7]:

- підтримка індексу, який завжди синхронізований з даними, що зберігаються в GraphDB;
- наявність кількох незалежних примірників для кожного сховища;
- повнотекстовий пошук з використанням нативних запитів Lucene;
- виділення фрагментів пошукових запитів в результаті пошуку;
- сортування за будь-яким попередньо настроєним полем;
- підкачка результатів з використанням offset і limit;
- вибір аналізатора;
- видалення тегів HTML / XML в літералах тощо.

Після етапу інтеграції онтології в RDF-сховище переходимо до створення внутрішньої архітектури інформаційної системи, яка використовує онтологічне сховище. Головними вимогами при побудові архітектури додатку є забезпечення доступу до бази знань. На рис. 9 показана загальна модель архітектури інформаційної системи валідації результатів неформального навчання. Основними користувачами системи є роботодавці, пошукувачі роботи та експерти. Взаємодія користувачів із системою (доступ до RDF-сховища) відбувається за допомогою веб-сервера, який буде обробляти запити до сховища, і обробляти результати запитів.

Для створення веб-серверу можуть застосовуватися засоби PHP та Java-Script. Порівняльний аналіз даних програмних засобів наведений у табл. 1.

Таблиця 1. Порівняльний аналіз програмних засобів PHP та Java-Script

Особливості	PHP	Java-Script
Мов скриптів на стороні сервера	Так	Ні
Оновлення файлів на сервері	Так	Ні
Підтримка бази даних	Так	Ні
Продуктивність	Повільний	Швидкий
Підтримка функцій	Більше	Менше

Головним фактором мови PHP є практичність. Мова PHP надає програмісту інструменти для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками:

- традиційністю;
- простотою;
- ефективністю;
- безпекою;
- гнучкістю.

У порівнянні з мовою PHP мова Java Script має такі особливості:

- знижений рівень безпеки через вільний доступ до вихідного коду популярних скриптів;
- безліч дрібних помилок на кожному етапі роботи. Велика частина з них легко виправляється, але їх наявність дозволяє вважати цю мову менш професійним, порівняно з іншими.

Таким чином, оптимальною з позиції швидкості розробки і безпеки середовищем для нашого проекту, є мова PHP.

Для розробки веб-додатків розглянемо PHP фреймворки Laravel і Symfony. В таблиці 2 наведені основні особливості цих фреймворків.

Фреймворк Symfony надає веб-розробникам вбудований функціонал для тестування, а також можливість працювати з компонентами і багаторазово використовувати код. Symfony 4.2 завантажує REST API за 2 мс. Це робить Symfony найшвидшим PHP-фреймворком. Symfony пристосовується до будь-яких вимог проекту. У цьому фреймворку є Event Dispatcher, завдяки якому Symfony є повністю конфігурованим. Крім того, всі компоненти Symfony управляються незалежно один від одного.

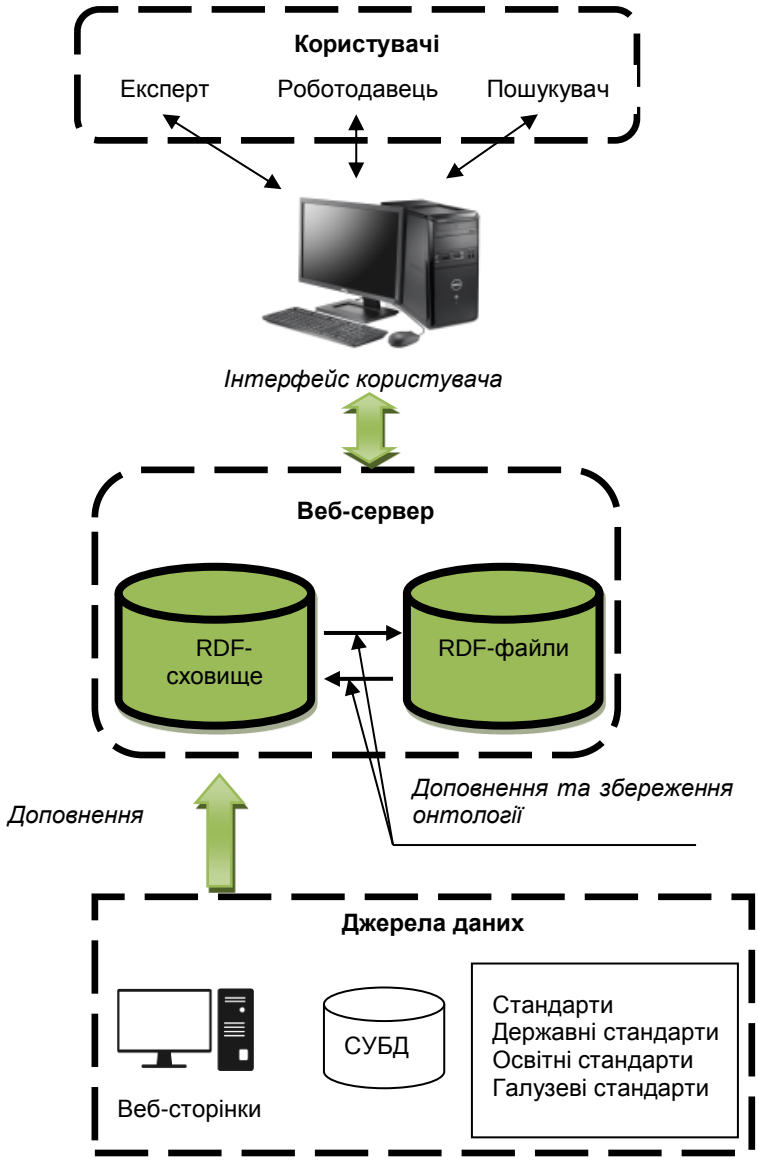

Рис. 9. Загальна архітектура додатку

Таблиця 2. Порівняльний аналіз програмних засобів Laravel і Symfony

Особливості	Symfony	Laravel
Кількість пакетів	7500	16 900
MVC	+	+
HTML-шаблони	Гілочка	Лезо
Установка через композитор	+	+
Повний текст пошуку	+ (ElasticSearch)	+ (ElasticSearch)
Підтримка CI, QA	-	PHPUnit

Головна мета Laravel – створення багатих функціоналом веб-додатків. Laravel має хороший движок шаблонів, що виконує велику кількість звичайних завдань.

В даному проекті доцільно використовувати фреймворк Laravel через його підвищену безпеку і готовність до установки плагінів і бібліотек. Цей фреймворк володіє таким функціоналом як RESTful-роутинг, кешування, управління користувачами і аутентифікація. Завдяки всьому цьому Laravel прискорює процес розробки.

На четвертому етапі для забезпечення діалогу користувача з інформаційною системою необхідна розробка інтерфейсу користувача. Найбільш актуальними для вирішення поставлених задач є React, Angular, Vue.js [14,15].

React − це відкрита JavaScript бібліотека для створення інтерфейсів користувача, яка покликана вирішувати проблеми часткового оновлення вмісту веб-сторінки, з якими стикаються в розробці

односторінкових застосунків. React дозволяє розробникам створювати великі веб-застосунки, які використовують дані, які змінюються з часом, без перезавантаження сторінки. Фреймворк React володіє такими перевагами: швидкість, простота, масштабованість. Концепція React побудована на інтерфейсі користувача на окремі самодостатні частини − компоненти, які досить просто підтримувати і розширювати.

Angular − написаний на TypeScript front-end фреймворк з відкритим кодом, який розробляється під керівництвом Angular Team у компанії Google, а також спільнотою приватних розробників та корпорацій [14]. Фреймворк Angular підтримує MVW концепцію і має такі переваги: швидке написання коду, швидке тестування будь-якої частини програми та двостороння прив'язка даних.

Vue.js − JavaScript-фреймворк, що використовує шаблон MVVM для створення інтерфейсів користувача на основі моделей даних, через реактивне зв'язування даних [15]. Vue.js реалізовує двосторонню прив'язку даних, візуалізацію на стороні сервера, Vue-cli (scaffolding інструмент для швидкого старту) і опціональну підтримку JSX.

Зважаючи на характеристики проаналізованих програмних засобів для розробки інтерфейсу користувача інформаційної системи приймаємо фреймворк React завдяки його ефективній роботі з інструментами обраними на попередніх етапах.

Інтерфейс користувача інформаційної системи валідації результатів неформального навчання складається з функціональної панелі та робочого вікна (рис. 10). Робоча форма пропонує користувачеві здійснити пошук за заданими параметрами: назвою професії, основними і додатковими навичками.

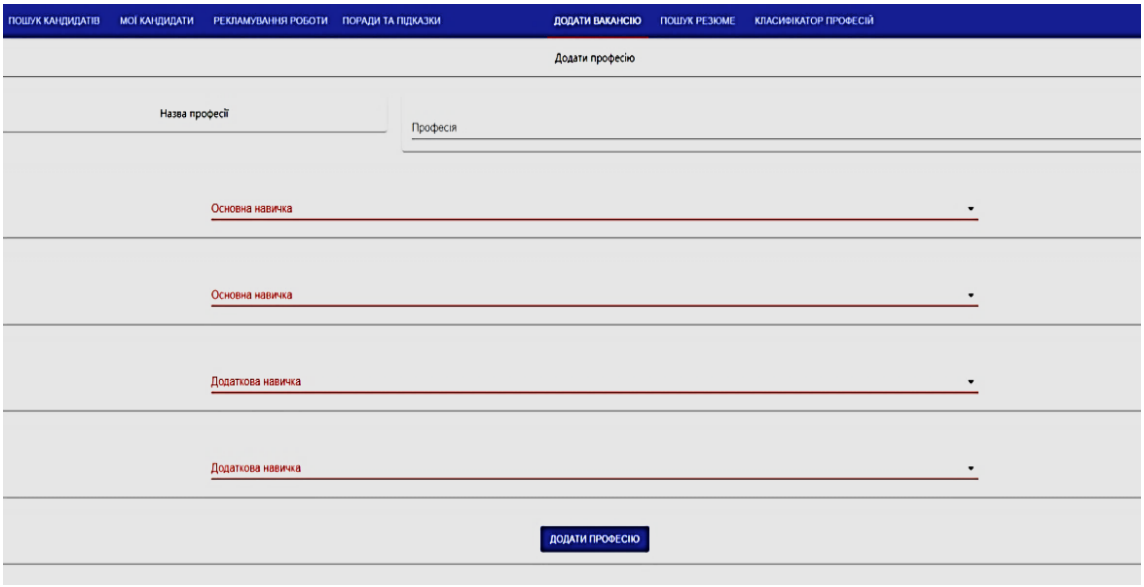

Рис. 10. Інтерфейс інформаційної системи

Перелік параметрів, за якими здійснюється пошук, вказується в електронному портфоліо користувача. Електронне портфоліо – це «візитівка» пошукувача, що містить дані про різні аспекти його діяльності, професійний розвиток, навчальну діяльність та персональні дані. Перелік параметрів, для додавання зберігається в онтології і може бути доповнений.

Також користувач має змогу за допомогою розробленої форми здійснити додавання нової професії за заданими параметрами. Для додання нової професії користувач вводить назву професії, вибирає основні та додаткові навички і натискає додати професію. Перелік параметрів, за якими здійснюється пошук навичок, зберігається в онтології. В результаті генерується запит до конектора через, який вносяться зміни до онтології.

Висновки

Аналіз розвитку сучасних інформаційних систем, заснованих на знаннях, показав доцільність використання ініціативи Semantic Web при розробці програмних засобів, призначенням яких є поєднання ринку освітніх послуг з ринком праці. Основними особливостями технології Semantic Web є розширення машинної обробки інформації, а саме інтелектуальне оброблення, засноване на семантичних технологіях. В роботі запропонована інформаційна система валідації результатів неформального навчання, основою якої виступає онтологія багатомовного класифікатора навичок ESCO. Для визначення основних функціональних можливостей інформаційної системи виконано функціональне моделювання IDEF0, декомпозиція якої виділила чотири блоки: проектування технічного завдання та архітектури системи; проектування бази даних системи; проектування моделі RDF; проектування фізичної моделі GraphDB. В статті описані розробки інформаційної системи валідації результатів неформального навчання, які включають створення схеми

онтології, процес інтеграції отриманої онтології в RDF-сховище, розробка архітектури додатку та створення інтерфейсу користувача системи. Для реалізації заявлених етапів розглянуті існуючі методи та програмні засоби представлення та аналізу знань, та їх придатність для розв'язання поставленої задачі, зокрема проаналізована система управління базами даних Neo4j, що обслуговує графову базу даних GraphDB; робота конекторів та запитів SPARQL до даних, розміщених в RDF-сховищі; охарактеризовані засоби створення веб-сервера, виконаний порівняльний аналіз PHP фреймворків для розробки веб-додатку, який за такими критеріями як безпека, готовність до установки плагінів та бібліотек, підтримка концепції MVC (модель–вигляд–контролер) виявив переваги програмного засобу Laravel. Для розробки інтерфейсу користувача інформаційної системи прийнятий фреймворк React завдяки його ефективній роботі з обраними на попередніх етапах інструментами.

Література

1. Прийма С.М., Рогушина Ю.В. Семантична обробка інформаційних ресурсів ринку праці. *Інформаційні технології і засоби навчання*. 2018. Т. 65. № 3. С. 337–355.
2. Sheila A. McIlraith, Tran Cao Son, and Honglei Zen. Semantic Web Services, Stanford University. URL: http://www.ksl.stanford.edu (дата звернення: 26.02.2020).
3. Pryima S., Rogushina J. Development of methods for support of qualification frameworks transparency based on semantic technologies. *Information Technologies and Learning Tools*. 2017. Vol. 59. N 3. P. 201–210.
4. Pryima S., Rogushina J. Ontological approach to qualifications matching on base of competences: model and methods. *Науковий вісник НГУ*. 2017. N 6. P. 162–168.
5. Davies J., Fensel D., van Harmelen F. Towards the Semantic Web: Ontology-driven knowledge management. *John Wiley & Sons Ltd*, England. 2002. 288 p.
6. Rogushina J., Priyma S. Use of competence ontological model for matching of qualifications. *Chemistry: Bulgarian Journal of Science Education*. 2017. Vol. 26. N 2. P. 216–228.
7. Pryima S.M., Rohushyna Yu.V., Strokan' O.V. Use of semantic technologies in the process of recognizing the outcomes of non-formal and informal learning. *CEUR Workshop Proceedings*. 2018. Vol. 2139. P. 226–235.
8. Зарипов А.А., Тузовский А.Ф. Разработка посистемы хранения метаданных семантической информационной системы *Институт кибернетики Национального исследовательского Томского политихнического института*. Томск, 2010. С. 53–58.
9. Horrocks I., Patel-Schneider P., van Harmelen F. From SHIQ and RDF to OWL: The making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web*. 2003. P. 7–26.
10. OWL Web Ontology Language. Overview. W3C Recommendation: W3C, 2009. URL: http://www.w3.org/TR/owl-features/ (дата звернення: 22.02.2020).
11. OWL 2 Web Ontology Language Document Overview. W3C. 2009. URL: http://www.w3.org/TR/owl2-overview/ (дата звернення: 24.02.2020).
12. Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web URL:http://www.sciam.com/article.cfm?articleid=000A0919 (дата звернення: 26.02.20. 20).
13. Шевченко Е. Л., Шевченко А. Ю., А. Н. Богдан Особенности построения прикладных программных систем на основе онтологических баз знаний. *АСУ и приборы автоматики : всеукр. межвед. науч.-техн. сб.* Харьков: Изд-во ХНУРЭ. 2011. Вып. 157. С. 36–41.
14. Angular.URL:https://uk.wikipedia.org/wiki/Angular_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) (дата звернення: 22.02.2020).
15. Vue.js. URL: https://uk.wikipedia.org/wiki/Vue.js (дата звернення: 22.02.2020).

References

1. Pryima S. & Rogushina J. Semantic processing of information resources in the market. Information technology and training tools. 2018. Vol. 65. N 3. P. 337–355. (in Ukrainian).
2. Sheila A. & Mcilraith Tran Cao Son & Honglei Zen. Semantic Web Services, Stanford University. URL: http://www.ksl.stanford.edu (дата звернення: 26.02.2020).
3. Pryima S., Rogushina J. Development of methods for support of qualification frameworks transparency based on semantic technologies. *Information Technologies and Learning Tools*. 2017. Vol. 59. N 3. P. 201–210. (in Ukrainian).
4. Pryima S., Rogushina J. Ontological approach to qualifications matching on base of competences: model and methods. *Науковий вісник НГУ*. 2017. N 6. P. 162–168. Available from: http://www.nvngu.in.ua/jdownloads/pdf/2017/06/06_2017_Rogushina.pdf
5. Davies J. & Fensel D. & van Harmelen F. Towards the Semantic Web: Ontology-driven knowledge management . John Wiley & Sons Ltd, England. 2002. 288 p. (in England).
6. Rogushina J., Priyma S. Use of competence ontological model for matching of qualifications. *Chemistry: Bulgarian Journal of Science Education*. 2017. Vol. 26. N 2. P. 216–228. (in Bulgaria)
7. Pryima S.M., Rohushyna Yu.V., Strokan' O.V. Use of semantic technologies in the process of recognizing the outcomes of non-formal and informal learning. *CEUR Workshop Proceedings*. 2018. Vol. 2139. P. 226–235. (in Ukrainian). Available from: http://ceur-ws.org/Vol-2139/226-235.pdf
8. Zaripov A. & Tuzovskiy A. Development of a metadata storage system for semantic information system. Institute of Cybernetics of National Research Tomsk Polytechnic Institute. Tomsk, 2010. P. 53–58. in Russian).
9. Horrocks I., Patel-Schneider P., van Harmelen F. From SHIQ and RDF to OWL: The making of a Web Ontology Language. *Web Semantics: Science, Services and Agents on the World Wide Web*. 2003. P. 7–26.
10. OWL Web Ontology Language. Overview. W3C Recommendation: W3C, 2009. URL: http://www.w3.org/TR/owl-features/ (дата звернення: 22.02.2020).
11. OWL 2 Web Ontology Language Document Overview. W3C. 2009. URL: http://www.w3.org/TR/owl2-overview/ (дата звернення: 24.02.2020).
12. Tim Berners-Lee, James Hendler and Ora Lassila. The Semantic Web URL:http://www.sciam.com/article.cfm?articleid=000A0919 (дата звернення: 26.02.20. 20).

13. Shenchenko YE. & Shenchenko A. & Bogdan. Features of the construction of applied software systems based on ontological knowledge bases. ACS and automation devices: all-in. inter. scientific and technical Sat. Kharkiv: Publishing House of KNURE. 2011. Vol. 157. P. 36–41. (in Ukrainian).
14. Angular.URL:https://uk.wikipedia.org/wiki/Angular_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA) (дата звернення: 22.02.2020).
15. Vue.js. URL: https://uk.wikipedia.org/wiki/Vue.js (дата звернення: 22.02.2020).

Одержано 03.02.2020

Про авторів:

Прийма Сергій Миколайович,
доктор педагогічних наук, професор,
професор кафедри комп'ютерних наук.
Кількість наукових публікацій в українських виданнях – 43.
Кількість наукових публікацій в зарубіжних виданнях – 15.
http://orcid.org/0000-0002-2654-5610,

Строкань Оксана Вікторівна,
кандидат технічних наук, доцент,
доцент кафедри комп'ютерних наук.
Кількість наукових публікацій в українських виданнях – 20.
http://orcid.org/0000-0002-6937-3548,

Рогушина Юлія Віталіївна,
кандидат фіз.-мат. наук,
старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 150.
Кількість наукових публікацій в зарубіжних виданнях – 31.
ORCID http://orcid.org/0000-0001-7958-2557,

Гладун Анатолій Ясонович,
кандидат технічних наук, доцент,
старший науковий співробітник відділу комплексних досліджень інформаційних технологій
Міжнародного науково-навчального центру інформаційних технологій та систем НАН та МОН України.
Кількість наукових публікацій в українських виданнях – 67.
Кількість наукових публікацій в зарубіжних виданнях – 53.
https://orcid.org/0000-0002-4133-8169,

Мозговенко Андрій Андрійович,
асистент кафедри комп'ютерних наук.
Кількість наукових публікацій в українських виданнях – 6.
https://orcid.org/0000-0002-7445-8925.

Місце роботи авторів:

Таврійський державний агротехнологічний університет імені Дмитра Моторного,
Мелітополь, Україна, проспект Богдана Хмельницького, 18.
Тел.: +38 (061) 42-20-32,

Інститут програмних систем НАН України
Київ, Україна, проспект Академіка Глушкова, 40, корп. 5.
Тел.: +38 (044) 526-33-19,

Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України,
03680, Київ, Україна, вул. Академіка Глушкова, 40,
Тел.: +38(044) 526-2549.
E-mail: pryima.serhii@gmail.com,
oksana.strokan@tsatu.edu.ua,
ladamandraka2010@gmail.com,
glanat@yahoo.com,
andrewmozg@gmail.com