

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»**

ННК «Інститут прикладного системного аналізу»
(повна назва інституту/факультету)

Кафедра Системного проектування
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ А.І.Петренко
(підпис) (ініціали, прізвище)

“ ___ ” _____ 2015 р.

Дипломна робота

першого (бакалаврського) _____ рівня вищої освіти

зі спеціальності 6.050101 «Комп’ютерні науки»
(код та назва спеціальності)

на тему: «Використання ресурсів Linked Open Data при перегляді веб сторінок»

Виконав (-ла): студент (-ка) 4 курсу, групи ДА-12
(шифр групи)

Королевський Герман Дмитрович _____
(прізвище, ім’я, по батькові) (підпис)

Керівник к.т.н,ст.в, Булах Б. В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант Охорона праці доцент Гусєв А.М. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Нормоконтроль _____ ст.. викладач Бритов О.А. _____

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2015 року

3. Архітектура плагіна
4. Алгоритми NLP
5. Реалізація та приклади роботи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо)

1. Технології Linked Open Data - плакат
2. Архітектура семантичного плагіна - плакат
3. Схема побудови запитів до бази знань - плакат
4. Реалізація семантичного плагіна - плакат

6. Консультанти розділів проекту (роботи)*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Гусев А. М., к.б.н., доцент		

7. Дата видачі завдання 01.02.2015

Календарний план

№ з/п	Назва етапів виконання дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1	Отримання завдання	01.02.2015	
2	Збір інформації	15.02.2015	
3	Вивчення варіантів реалізації та вибір варіанту для розробки	28.02.2015	
4	Розробка алгоритму та архітектури	10.03.2015	
5	Розробка плану тестування	15.03.2015	
6	Розробка програмної моделі	25.03.2015	
7	Розробка опису плагіну	25.04.2015	
8	Тестування програми	30.04.2015	
9	Оформлення дипломної роботи	31.05.2015	
10	Отримання допуску до захисту та подача роботи в ДЕК	18.06.2015	

Студент

(підпис)

Королевський Г.Д.

(ініціали, прізвище)

Керівник проекту (роботи)

(підпис)

Булах Б.В.

(ініціали, прізвище)

АНОТАЦІЯ

бакалаврської дипломної роботи Королевського Германа Дмитровича
на тему: «Використання ресурсів Linked Open Data при перегляді веб сторінок»

Дипломна робота присвячена дослідженню семантичних ресурсів Open Linked Data та візуалізації доступу до баз знань, які входять до цих ресурсів. В роботі також є аналіз існуючих технологій доступу до баз знань та їх реалізацій у семантичному веб. Досліджені методи перетворення natural language пошуків у SPARQL запит, що базуються на словнику підтримуваних слів. Запропонована концепція плагіна для веб браузеру та його архітектура, як рішення питання візуалізації запитів та графічного інтерфейсу користувача. Результатом роботи став програмний продукт (плагін), що дозволяє здійснювати пошук по семантичним ресурсам Open Linked Data, перетворюючи звичайну людську мову на запит SPARQL. Тези дипломної роботи були представлені на конференції «Системний аналіз та інформаційні технології - 2015». Крім зручної візуалізації запитів до ресурсів Open Linked Data, алгоритми та концепції представлені в цій роботі також можна використовувати для побудови систем питання-відповідь, пакетах для створення та обслуговування семантичної розмітки на веб порталах.

Загальний об'єм роботи: 76 сторінок, 25 рисунків, 9 таблиць, 14 посилань.

Ключові слова: семантичний веб, SPARQL, Open Linked Data, графічний інтерфейс, візуалізація запитів, NLP.

АННОТАЦИЯ

бакалаврской дипломной работы Королевского Германа Дмитриевича
на тему: «Использование ресурсов Linked Open Data при просмотре веб страниц»

Дипломная работа посвящена исследованию семантических ресурсов Open Linked Data и визуализации доступа к базам знаний, входящих в эти ресурсы. В работе также есть анализ существующих технологий доступа к базам знаний и их реализаций в семантическом веб. Исследованы методы преобразования natural language поисков в SPARQL запрос, основанные на словаре поддерживаемых слов. Предложена концепция плагина для веб браузера и его архитектура, как решение вопроса визуализации запросов и графического интерфейса пользователя. Результатом работы стал программный продукт (плагин), который позволяет осуществлять поиск по семантическим ресурсам Open Linked Data, превращая обычную человеческую речь в запрос SPARQL. Тезисы дипломной работы были представлены на конференции «Системный анализ и информационные технологии - 2015». Кроме удобной визуализации запросов к ресурсам Open Linked Data, алгоритмы и концепции, представленные в этой работе можно использовать для построения вопрос-ответных систем, пакетах для создания и обслуживания семантической разметки веб порталов.

Общий объем работы: 76 страниц 25 рисунков, 9 таблиц, 14 ссылок.

Ключевые слова: семантический веб, SPARQL, Open Linked Data, графический интерфейс, визуализация запросов, NLP.

ABSTRACT

Korolevsky Herman bachelor's degree thesis: "Surfing on the web using Linked Open Data resources"

This thesis is devoted to research of Open Linked Data semantic resources and access visualization to knowledge bases belonging to these resources. Also provided analyzes of the existing knowledge base access technologies and their implementations in the semantic web. Studied natural language algorithms, which convert NL search query into SPARQL query based on supported dictionary words. The concept of plug-in for web browser and its architecture was introduced as a solution to visualize queries and graphical user interface. The final result is a software (plug-in), which allows user to query semantic resources of Open Linked Data, turning ordinary human speech to a SPARQL query. Abstracts of this thesis were presented at the conference "System analysis and information technologies - 2015". In addition to convenient Open Linked Data query visualization, algorithms and concepts presented in this paper can be used to develop a question-answer systems, and packages for creating and maintaining semantic markup in web portals.

Total volume: 76 pages, 25 figures, 9 tables, 14 references.

Keywords: Semantic Web, SPARQL, Open Linked Data, graphical user interface, visualization requests, NLP

ЗМІСТ

Перелік умовних позначень, символів, скорочень і термінів.....	9
ВСТУП	10
1. БАЗОВІ ПРИНЦИПИ ТА ТЕХНОЛОГІЇ LINKED OPEN DATA	12
1.1. Огляд зв'язаних даних (<i>Open Linked Data</i>).....	12
1.2 Технології та стандарти <i>Linked Open Data</i>	14
1.3. Аналіз та порівняння існуючих баз знань.	21
1.3.1. Порівняння баз знань	23
1.4. Висновок	27
2. ПРАКТИЧНА ЧАСТИНА	29
2.1. Робота з базою знань DBpedia	29
2.2. Концепція семантичного плагіна Semantic Quick Search (SQS).....	35
2.3. Розробка розширення. Доцільність проекту	38
2.3.1. Вимоги та технічне завдання до розширення Semantic Quick Search (SQS).....	38
2.3.2. Засоби розробки SQS	39
2.3.3. Доцільність концепції.....	42
2.4. Аналіз тексту. Алгоритми NLP.....	43
2.5 Архітектура SQS	45
2.6 Висновок	49
3. РЕАЛІЗАЦІЯ ТА ПРИКЛАДИ РОБОТИ	50
3.1 Реалізація.....	50
3.2 Приклади роботи програми SQS	53
3.3 Аналіз результатів роботи	57
3.4 Висновок	60
4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	61
4.1. Вступ та теоретичні відомості охорони праці.....	61
4.2. Загальні вимоги до приміщення	62
4.3. Вимоги до організації робочого місця	63
4.4. Аналіз параметрів робочого місця та приміщення.....	64

4.4.1. Геометричні параметри приміщення	64
4.4.2. Параметри робочого місця	65
4.4.3. Аналіз та розрахунок стану освітлення в приміщенні	66
4.4.5. Виробничий шуму й вібрації	70
4.4.6. Вимоги до пожежної та електро безпеки.....	71
4.5. Висновок	72
ВИСНОВКИ.....	73
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75

Перелік умовних позначень, символів, скорочень і термінів

Скорочення:

- ПЗ – програмне забезпечення;
- SQS – Semantic Quick Search, назва плагіну;
- LOD – Linked Open Data, концепція семантично зв'язаних даних;
- SPARQL - Protocol and RDF Query Language, мова запитів до ресурсів LOD;
- MQL – Metaweb query language, API для доступу до Freebase;
- NLP – Natural language processing, напрям комп'ютерних наук, що вивчає програмну обробку природньої мови;
- RDF – Resource Description Framework;
- URI – Uniform Resource Identifier;
- OWL – Web Ontology Language;

Терміни:

- Семантичний веб - це надбудова над сучасною Всесвітньою павутиною, яка покликана зробити інформацію, що розміщена в мережі, зрозумілішою для комп'ютерів;
- Триплет – в світі семантичного веб відношення типу: суб'єкт, предикат, об'єкт.
- База знань - це особливого роду база даних, розроблена для управління знаннями (метаданими): збором, зберіганням, пошуком і видачею знань.
Розширення (плагін) для браузеру – програмний модуль, надбудова для браузеру, що дозволяє розширити стандартний функціонал.

ВСТУП

Пошук інформації – саме так на сьогоднішній день можна охарактеризувати тенденції розвитку інформаційних технологій, які диктуються потребами користувачів. Так передові компанії створили Google Search, Microsoft Bing, Yandex Поиск та ін. Ці системи в цілому задовольняють потреби користувача, але результатом пошуку є набір найбільш релевантних посилань на сайти, на яких згадуються шукані ключові слова. Отже, ці системи не дають однозначної відповіді на запит – користувач самостійно проводить подальший аналіз отриманих даних, витрачаючи власний час.

Комп'ютери беруть досить обмежену участь у формуванні й обробці інформації в мережі Інтернет. Функції комп'ютерів в основному зводяться до збереження, відображення і пошуку інформації. У той же час створення інформації, її оцінка, класифікація й актуалізація — усе це як і раніше виконує людина. Потрібно використовувати мову, що була б зрозумілою комп'ютеру. Тобто, вся інформація в Інтернеті повинна розміщуватись двома мовами: людською мовою для людини і комп'ютерною мовою для розуміння комп'ютера. На початку XXI століття Сер Тім Бернерс-Лі вперше запровадив термін *Semantic Web (Web 3.0)* та поняття *Open Linked Data*, описав нові стандарти [1]. Семантична павутина — це концепція мережі, у якій кожен ресурс людською мовою був би доповнений описом, зрозумілим комп'ютеру.

На сучасному етапі розвитку в світі семантичного веб нереалізованим залишається графічний інтерфейс користувача, оснований на обробці *natural language* запитів, які перетворюються на *SPARQL* запит. В цілому ці компоненти утворюють шар, що дозволить людині вільно використовувати нову технологію.

Мета дипломної роботи – проаналізувати сучасний стан концепції *Open Linked Data*, особливо підходи до реалізації графічних інтерфейсів користувача, та розробити власний програмний продукт, який дає можливість доступно робити запити до ресурсів *Open Linked Data*. Використати алгоритми NLP для перетворення людської мови у SPARQL запит.

Для досягнення мети були посалені наступні завдання:

- Зробити огляд стеку технологій семантичного веб,
- Проаналізувати існуючий інструментарій та визначити чого не вистачає для коректної роботи з семантичним веб,
- Порівняти найбільші бази знань та способи доступу до них,
- Описати концепцію програмного продукту, що буде реалізовано,
- Аналіз алгоритмів *NLP* для перетворення природньої мови у SPARQL запит.

Об’єкт дослідження – ресурси *Open Linked Data* та технології стеку

Предмет дослідження – візуалізація запитів до ресурсів *Open Linked Data*.

1. БАЗОВІ ПРИНЦИПИ ТА ТЕХНОЛОГІЇ LINKED OPEN DATA

1.1. Огляд зв'язаних даних (*Open Linked Data*)

Linked Open Data – це живий приклад семантичної мережі, колекція взаємопов'язаних даних у мережі, надбудова над сучасною Всесвітньою павутиною, яка покликана зробити інформацію, що розміщена в мережі, такою, що може бути оброблена комп'ютером. Відомо, що майже вся інформація в Інтернеті знаходиться в текстовій формі. Не секрет також, що прогрес в галузі обробки людської мови (англ. Natural Language Processing, NLP) йде повільно. Комп'ютери не можуть сприйняти й осмислити словесну інформацію, розміщену в Інтернеті, і в найближчий час, не зможуть. Тоді постає задача — розробити систему, що дозволить комп'ютерам обробляти розміщену в мережі інформацію і дати можливість автоматично аналізувати її. Рішенням цієї проблеми є концепція семантичної павутини. Слово «семантична» у цьому випадку означає «осмислена», «зрозуміла».

Принципи Linked Open Data по Тіму Бернерсу-Лі:

1. Використовувати URI, щоб назвати (визначити) поняття.
2. Використовувати HTTP, URI, так щоб ці поняття можна було інтерпретувати програмно.
3. Надати корисну інформацію про те, що ім'я позначає, коли воно інтерпретується, використовуючи відкриті стандарти, такі як RDF, SPARQL і т.д.
4. Зв'язати поняття з іншими, використовуючи їх HTTP URI, при публікації даних в Інтернеті.

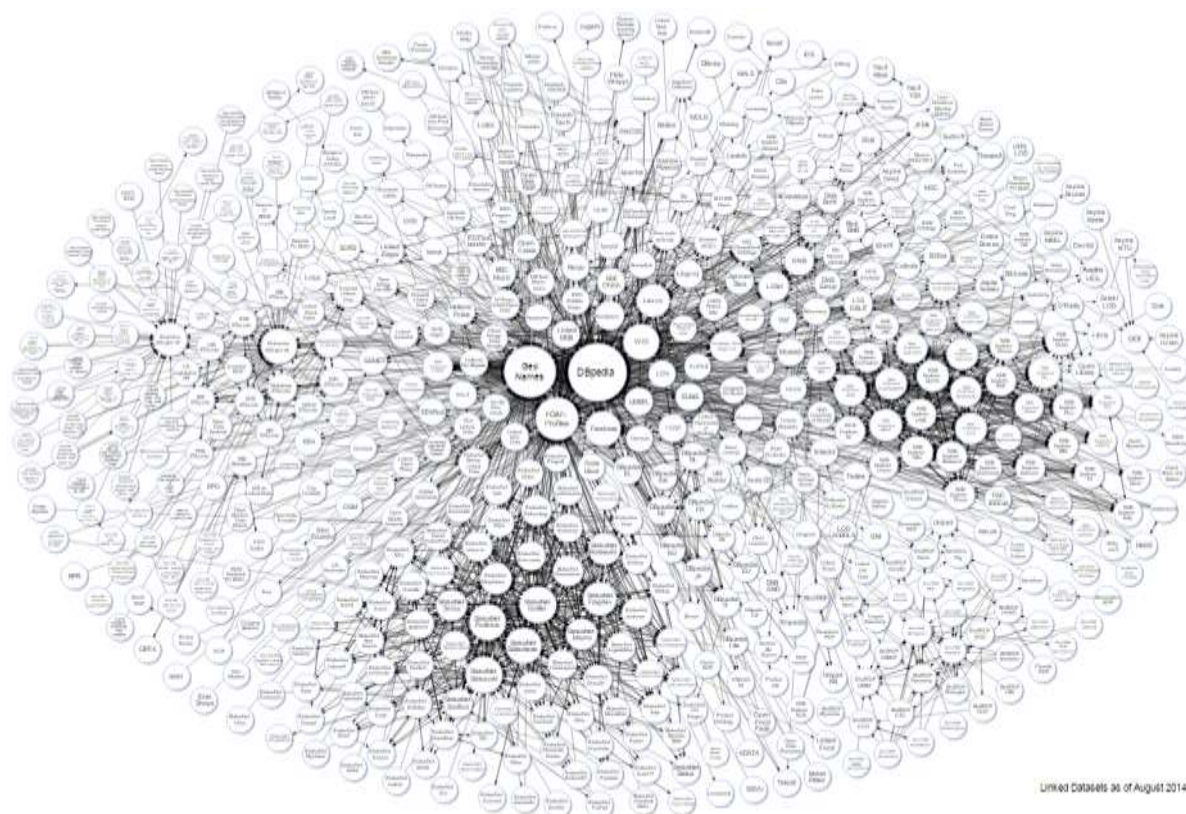


Рис.1 – Візуальне представлення структури LOD на 2014 рік

Вище показана схема зав'язків між базами знань LOD, яку будують кожен рік (рис. 1). Ядром структури є DBpedia – база знань, яка автоматично витягує структуровану інформацію з Wikipedia. Freebase - велика колаборативна база знань, яка містить метадані, зібрані, в основному, інтернет-спільнотою. Це онлайн-колекція структурованих даних, зібраних з багатьох джерел, наприклад, окремих вікі-проектів. Також слід згадати GeoNames - географічна база даних, має доступний вікі-інтерфейс. Всі структуровані бази знань називати немає сенсу, адже всі вони утворюють, в цілому, одну велику зв'язану базу знань, але в наступних розділах будуть більш конкретно описані найбільші бази знань, засоби доступу до них та інфраструктура.

Слід відмітити, що LOD будується на інформаційній моделі семантичної мережі, тобто всі дані представлені у вигляді орієнтованого графа, вершини якого відповідають об'єктам предметної області, а ребра задають відносини між ними (рис. 2). Об'єктами можуть бути поняття, події, властивості, процеси.

Таким чином, семантична мережа відображає семантику предметної області у вигляді понять і відносин між ними [3].

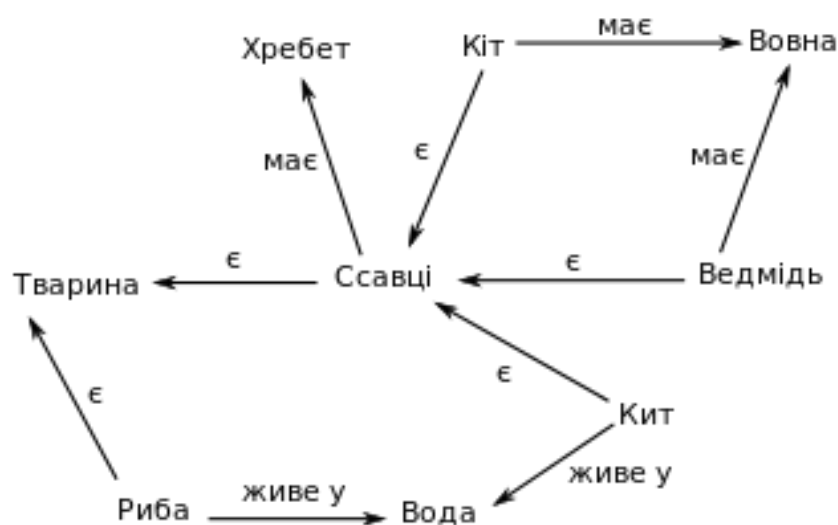


Рис. 2 – Граф зв'язаних даних

1.2 Технології та стандарти *Linked Open Data*

Семантичний веб працює на основі стандартизованого стеку, який містить в собі ієрархію мов та технологій розмітки (рис.2). Їх можна розділити на 3 уявних рівня: низький (технології звичайного веб), середній (мови та стандарти опису саме семантичного веб, які дозволяють будувати онтології) та елементи верхнього рівня (не до кінця розроблені концепції, які дозволять захистити інформацію та надати їй зручний для людини вигляд) [2].

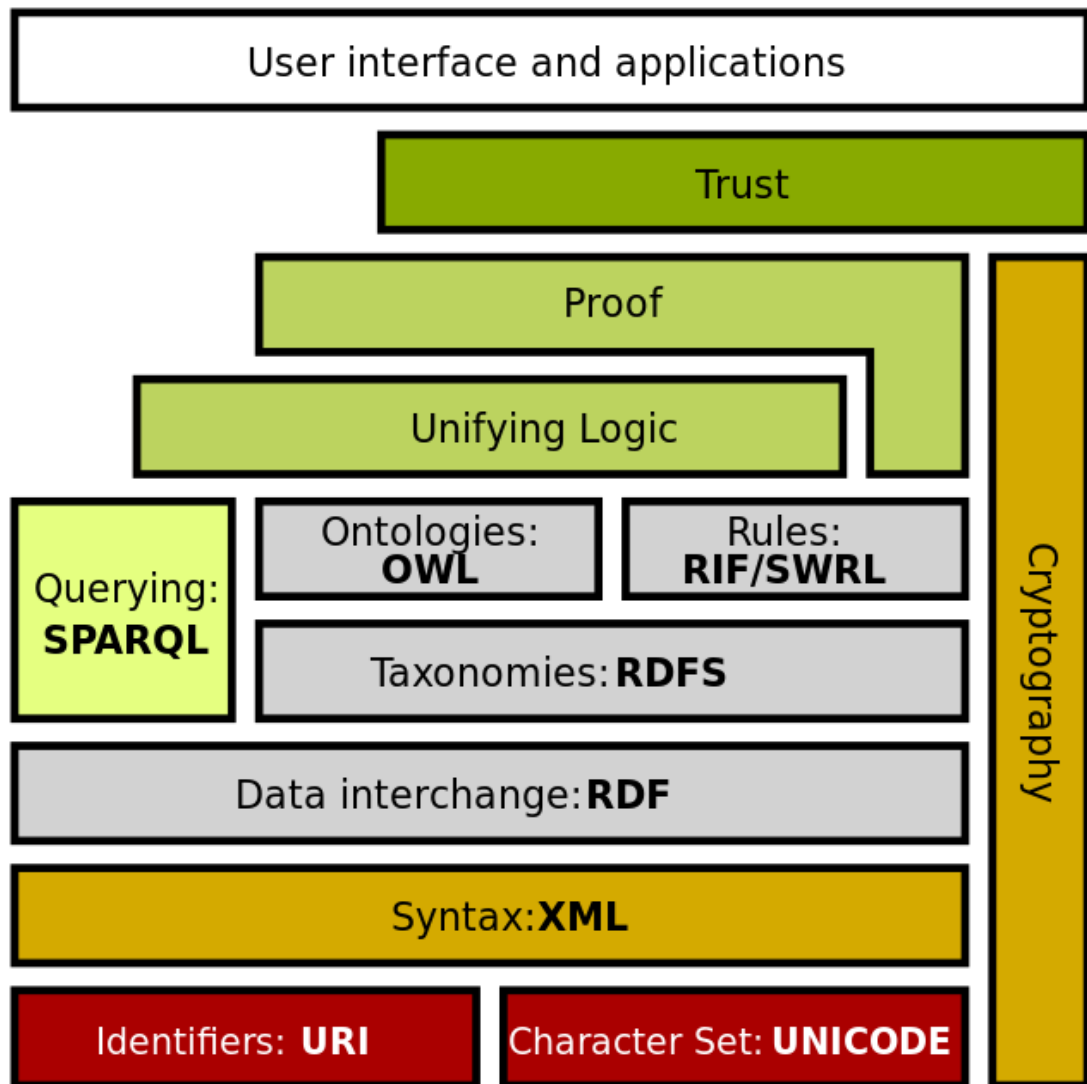


Рис. 3 – Стек семантичного веб

Функції та взаємозв'язки між шарами, починаючи з нижніх, можна описати наступним чином:

1. Hypertext Web technologies

Нижні шари містять технології, які добре відомі з гіпертекстової мережі і без змін створюють основу семантичної мережі.

IRI (*Internationalized Resource Identifier*), узагальнення URI (*uniform resource identifier*), унікальний ідентифікатор семантичних веб-ресурсів.

Семантична мережа потребує унікальну ідентифікацію, для коректної подальшої обробки інформації у верхніх шарах.

URI - це послідовність символів, що ідентифікує абстрактний або фізичний ресурс. При цьому URI може вказувати як місце розташування ресурсу (URL), так і його ім'я (URN). А може містити і те й інше. Тобто URL і URN - це окремі випадки URI.

URI будується за певними правилами і складається з обов'язкових схеми і ієрархічної частини, а також опціональних запиту (йому передує знак "?") і фрагмента (йому передує знак "#"). Ієрархічна частина в свою чергу складається з необов'язкового Authority та обов'язкового шляху. Authority включає в себе Userinfo (логін і пароль), хост і порт. Крім того, шлях може містити так звані параметри.

На схемі це виглядає ось так (рис. 4):

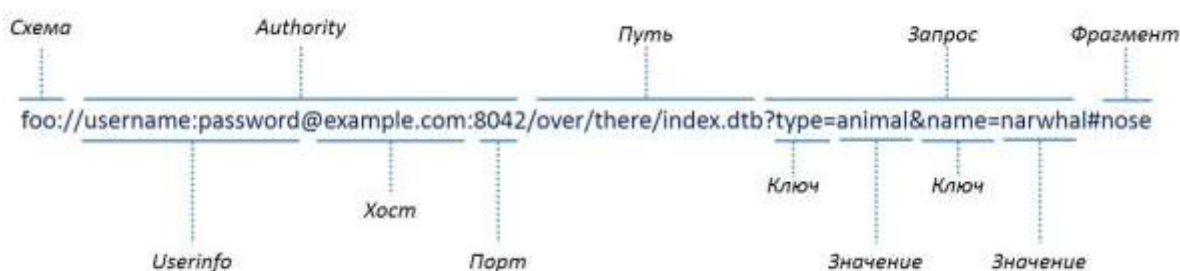


Рис. 4 – Схема структури ідентифікатору URI

Unicode служить для представлення і редагування тексту на різних мовах. Семантична мережа повинна також сприяти поширенню інформації на різних мовах.

XML є мовою розмітки, яка дозволяє створювати документи, що складаються з структурованих даних. Семантичний веб надає сенс (семантику) структурованим даним.

XML Namespaces дає можливість використовувати розмітку різних джерел, для зв'язування даних.

2. Стандартизовані технології семантичного веб

Це середній шар, містить стандартизовані технології W3C, дозволяє будувати семантичні веб-додатки. Нижче детально розглянуто всі стандарти.

Resource Description Framework (RDF) основа для створення твердження у формі так званих триплетів. Це дозволяє представляти інформацію про ресурси у вигляді графа (Рис. 5). Для створення зрозумілого комп'ютеру опису ресурсу в семантичній павутині використовується формат RDF (Resource Description Framework), що заснований на синтаксисі XML і використовує ідентифікатори URI для позначення ресурсів. RDF був затверджений як стандарт W3C у лютому 2004 року. RDF — це система опису мережних ресурсів, зрозуміла комп'ютеру. Формат RDF призначений для збереження метаданих (метадані — це дані про дані). Відповідно до концепції семантичної павутини, опис у форматі RDF повинен прикріплюватися до кожного мережного ресурсу. Документи RDF повинні оброблятися комп'ютером автоматично, RDF не призначений для читання і використання людиною. На сьогодні формат RDF вже сформувався й одержав широке поширення, він служить каркасом для створення семантичної павутини.

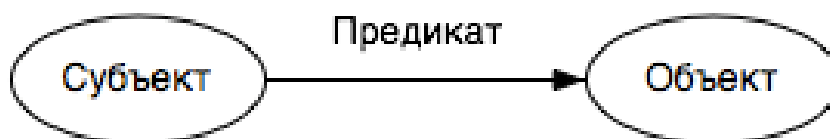


Рис.5 – Триплет опису відношення між поняттями RDF

Суб'єкт – описуваний ресурс.

Предикат - властивість ресурсу.

Об'єкт – значення властивості.

Приклад:

```
<?xml version="1.0"?>

<rdf:RDF

xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire_Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Columbia</cd:company>
  <cd:price>10.90</cd:price>
  <cd:year>1985</cd:year>
</rdf:Description>

</rdf:RDF>
```

Перший рядок документа RDF це декларація XML. Слідом за декларацією XML йде кореневий елемент RDF документів: <rdf: RDF>.

Простір імен xmlns: RDF, вказує, що елементи з префіксом rdf містяться в просторі імен "http://www.w3.org/1999/02/22-rdf-syntax-ns#".

Простір імен xmlns:cd, вказує, що елементи з префіксом cd містяться в просторі імен "HTTP: //www.recshop.fake/cd#".

Елемент <rdf: Description> містить опис ресурсу, ідентифікованого rdf:about.

Елементи: <cd: artist>, <cd: country>, <cd: company> і т.д., властивості ресурсу.

В результаті маємо детальний опис об'єкту Empire Burlesque, включаючи автора, рік випуску та іншими властивостями. Але це ще не семантика. Для того, щоб зв'язати поняття використовують семантичне розширення RDF – RDF schema.

Schema RDF (RDFS) забезпечує базовий словниковий запас для RDF. Використання RDFS дозволяє, наприклад, створити ієрархію класів та властивостей. Схема - набір класів і властивостей для моделі представлення знань RDF, що становить основу для опису онтологій з використанням розширеного RDF-словника для структури RDF-ресурсів. RDFS використовує кодування у вигляді RDF, тому триплети, що відносяться до RDF можуть зберігатися, оброблятися і запитуватися подібно описам RDF-ресурсів, наприклад, за допомогою SPARQL. Таким чином за допомогою схем можна побудувати відношення між класами і об'єктами, тобто семантично їх зв'язати.

Web Ontology Language (OWL) розширює RDFS, додає більше детальні конструкції для опису семантики RDF тверджень. Це дозволяє встановлювати додаткові обмеження, такі як, потужність множин, обмеження значень або характеристики властивостей (транзитивність). Якщо стандарт RDF оперує переважно інформацією, то OWL додає до RDF правила виводу нових фактів.

SPARQL це мова запитів RDF – використовується для посилення запиту до будь-яких даних на основі RDF. Мова запитів необхідна для отримання інформації для семантичних веб-додатків.

Загальна схема SPARQL-запиту виглядає так:

```

PREFIX foo: <http://example.com/resources/>
# префіксні оголошення
FROM ...
# джерела запиту
SELECT ...
# пункт результату
WHERE {...}

```

```
# критерії запиту
ORDER BY ...
# модифікатори запиту
```

Префіксні оголошення слугують для скорочення універсальних ідентифікаторів ресурсу.

Джерела запиту визначають, які RDF графи запитуються.

Пункт результату повертає набір даних (вибірку), які задовольняють умові.

Критерії запити визначає, що запитувати в базовому наборі даних

Модифікатори запити обмежують, упорядковують, і інакше перетворюють результати запити.

Існує досить багато реалізацій пошукових систем та іншого ПЗ на базі запитів SPARQL, але всі вони працюють на необробленому, сирому SPARQL запиті і тому ці сервіси за жодних обставин не будуть у широкому використанні. На сьогоднішній день, актуальною є розробка алгоритмів обробки запитів на людській мові, які потім перетворюються на запит до Open Linked Data ресурсів на SPARQL.

3. Нереалізовані технології семантичного веб

Верхні шари містять технології, які ще не стандартизовані або існують тільки у вигляді ідеї, яка повинна бути реалізована у семантичній мережі.

Криптографія потрібна для підтвердження, що джерело семантичних даних надійне. Слід ставити відповідні цифрові підписи на RDF твердженнях.

Графічний інтерфейс користувача - це останній шар, що дозволить людям використовувати семантичні веб-додатки. На даний момент немає універсального зручного підходу до створення такого інтерфейсу, який би дозволяв використовувати повний спектр можливостей семантичного веб.

У розділі 3 буде розглянута детальна концепція та реалізація семантичного плагіна для браузеру з алгоритмом аналізу тексту, який дозволить проводити пошук виділеного тексту на веб сторінці серед Linked Open Data.

Як висновок до розділу слід сказати, що вже існує надійна основа семантичного веб, яка включає старі, вже освоєні технології старого веб (URI, XML), а також унікальні та нові стандарти семантичного веб, тісно пов'язані між собою. Це RDF, OWL, SPARQL – розмітка для триплетів понять, опис онтологій, побудова запитів до баз знань. Всім цим вже користуються, але не вистачає ще декількох основних стандартів, що забезпечували б довір'я до джерел інформації, їх захищеність. Окремим пунктом треба виділити візуалізацію роботи з семантичним веб, тобто графічний інтерфейс – це слабе місце семантичної павутини і потребує детального пошуку оптимальних рішень.

1.3. Аналіз та порівняння існуючих баз знань.

Світ семантичного веб, а саме його підрозділ у вигляді Open Linked Data вже включає в себе 31,634,213,770 триплетів та близько 500 мільйонів взаємозв'язків між базами знань і ці цифри невпинно ростуть. LOD технології зараз використовують для обміну даними, що охоплюють широкий спектр областей знань. Наведена нижче таблиця (табл. 1) показує загальну кількість триплетів, кількість посилань на RDF бази, а також наглядна діаграма кількісного розподілення триплетів по областям (рис. 6). Приведені дані, взяті з офіційного сайту LOD станом на 2014 рік.

Таблиця 1 – Сумарна кількість даних LOD

Області	К-сть баз знань	Триплети	Зовнішні посилання
Медіа	25	1,841,852,061	50,440,705
Географія	31	6,145,532,484	35,812,328
Урядова ін-ція	49	13,315,009,400	19,343,519
Публікації	87	2,950,720,693	139,925,218
Змішана ін-ція	41	4,184,635,715	63,183,065
Природничі науки	41	3,036,336,004	191,844,090
Контент користувачів	20	134,127,413	3,449,143
Всього	295	31,634,213,770	503,998,829

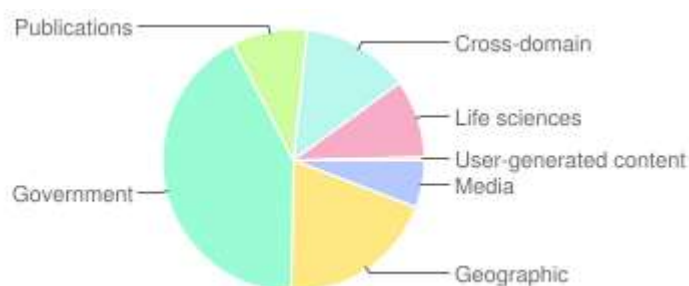


Рис. 6 – Діаграма кількісного розподілення триплетів за областями

З приведених даних видно, що об'єм LOD надзвичайно великий. Виходячи з концепції (детальніше буде розглянуто в розділі 3) пошукового плагіна немає сенсу використовувати як джерело знань всю LOD – на один пошуковий запит буде багато нерелевантної та продубльованої інформації, також це сильно загальмує пошук. Отже, основними джерелами даних для нас є міжнародні бази знань, що включають довідкові, лінгвістичні та енциклопедичні дані такі як: *DBpedia*, *Freebase*, *Wikidata* [4].

1.3.1. Порівняння баз знань

В цій частині будуть порівняні 3 найбільші бази знань: *DBpedia*, *Freebase*, *Wikidata*. Нас цікавлять їх об'єм, засоби доступу, API, розповсюдженість у готових програмних продуктах та ін.

DBpedia

DBpedia - співтовариство, зусилля якого спрямовані на те, щоб витягати структуровану інформацію з Wikipedia і робити цю інформацію доступною в Web. DBpedia дозволяє створювати уточнені запити до Wikipedia, і прив'язує інші набори даних у Web до даних Wikipedia.

Проект DBpedia підсилює гігантське джерело знання Wikipedia, розпаковуючи структуровану інформацію від Wikipedia і тим самим робить цю інформацію доступною в Web на основі GNU Free Documentation License.

База знань DBpedia описує більш ніж 3.4 мільйонів об'єктів, зокрема як мінімум 213,000 людей, 328,000 місць, 57,000 музичних альбомів, 36,000 фільмів, 20,000 компаній. База знань вміщує 274 мільйонів уривків інформації (RDF triples). Ярлики і короткі описи об'єктів 14-ма різними мовами; 609,000 зв'язків із зображеннями і 3,150,000 зв'язків із зовнішніми веб-сторінками; 4,878,000 зовнішніх посилань в інших RDF наборах даних, 415,000 категорії Wikipedia, і 75,000 категорії YAGO.

База знань DBpedia має декілька переваг над існуючими базами знань: охоплює багато областей; відображає реальну колективну згоду; автоматично еволюціонує як і Wikipedia, і вона справді багатомовна.

Статті Вікіпедії в основному складаються з суцільного тексту, але також включають в себе структуровану інформацію вбудовану в статті, таку, як в інфобоксах та таблицях (висувні панелі, які з'являються у верхньому правому

куті за замовчуванням багатьох статей Вікіпедії або на початку мобільні версії), класифікаційна інформація, зображення, координати і посилання на зовнішні веб-сторінки. Ця структурована інформація витягується у DBpedia та стає частиною універсального датасету, який можна дістати через запит.

Проект написаний на Java, Scala, VSP та має ліцензію GNU General Public License, що дозволяє їм вільно користуватися. На жаль, у проекту відсутнє API, доступ до даних відбувається тільки через SPARQL запити на кінцеву точку www.dbpedia.org/sparql. На основі даних з DBpedia вже реалізовано досить багато реальних проектів, серед яких:

1. Marbles Linked Data Browser

Marbles це серверний додаток, що форматує Семантичний веб-контенту для клієнтів, що використовують XHTML. Кольорові точки використовуються для відображення походження відображуваних даних зі списком джерел даних, звідси й назва.

Marbles забезпечує відображення можливостей для DBpedia Mobile. Скріншот роботи браузеру на рис. 7

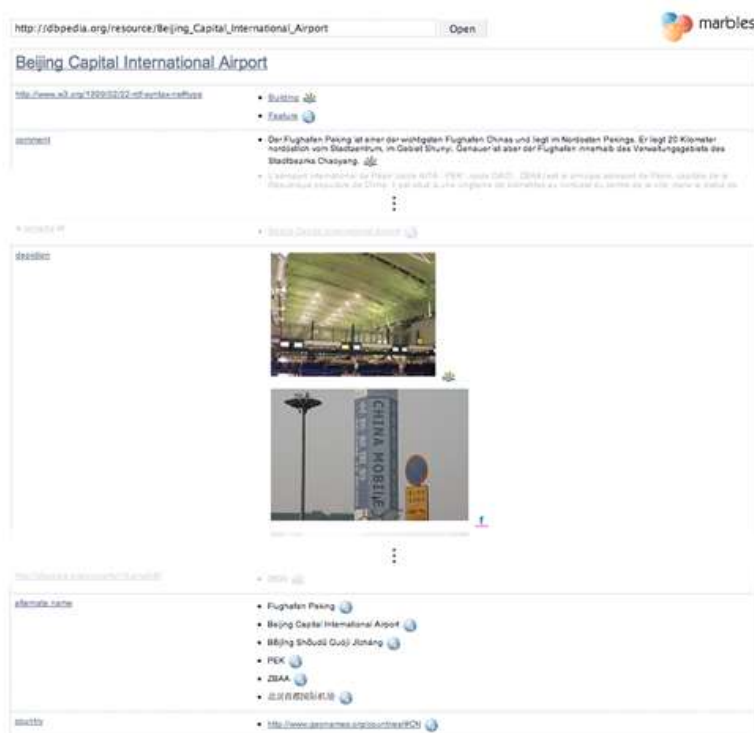


Рис. 7 – Робота браузеру Marbles

2. Query – фреймворк для python для перетворення запитів на природній мові питання на мову запитів до бази знань SPARQL, може бути легко налаштований на різні види питань природною мовою і запити до бази знань.

Freebase

Freebase - велика колаборативна база знань що містить метадані складені в основному спільнотою. Це онлайн колекція структурованих даних зібраних з багатьох джерел, наприклад окремих 'Вікі'. Freebase має на меті створити глобальний ресурс що дозволить людям (та машинам) отримувати доступ до загальновідомої інформації набагато ефективніше. Розробляється американською софтверною компанією Metaweb і працює публічно з березня 2007. Metaweb була куплена Google.

Запити до бази здійснюються мовою "Metaweb Query Language" (MQL). Дані Freebase доступні free/libre для комерційного та некомерційного використання під ліцензією Creative Commons Attribution. Для програмістів надається API, RDF endpoint, та дампи. Google News Timeline включає медіаінформацію з Freebase. Freebase має інтерфейс, який дозволяє непрограмістам заповнювати її інформацією, та категоризувати і об'єднувати елементи даних в змістовний, тобто семантичний спосіб. Freebase містить дані зібрані з таких джерел як Вікіпедія, ChefMoz, NNDB, та MusicBrainz, та дані внесені користувачами.

Реалізовані проекти:

1. Google Refine - інструмент для очищення та отримання даних
2. Powerset - семантичний пошуковий рушій, який здійснює пошук по Freebase відповідей на питання задані природньою мовою.
3. Freebase genealogy - переглядач родоводів.

4. FMDb - Freebase IMDB
5. Freebase sets - клон Google sets що використовує дані Freebase
6. Parallax data viewer - альтернативний інтерфейс користувача
7. Freebase Schema Explorer - візуалізатор онтологій Freebase
8. Thinkbase - візуальний інструмент що базується на графах

На жаль, проект Freebase згортають до червня 2015 року та переносять весь контент на новий проект Google Knowledge Graph, переписують API.

WikiData

Wikidata — загальне централізоване сховище для різних типів даних, таких як, інтервікі, метричні параметри, чисельність населення, та інша статистична інформація, що може бути використана на сторінках проектів фонду Wikimedia. Розробка проекту розпочата на Wikimedia Deutschland. Проект запущений 30 жовтня 2012 року, що дозволило йому стати першим новим проектом Wikimedia Foundation з 2006 року.

Дані в WikiData опублікована під Creative Commons Public Domain Dedication 1.0, що дозволяє повторне використання даних. Є можливість копіювати, змінювати, розповсюджувати, навіть в комерційних цілях, не питаючи дозволу. Дані в WikiData вводяться і підтримується редакцією WikiData.

Редагування, користування, перегляд даних повністю багатомовне. Дані, введені на будь-якій мові відразу доступні у всіх інших мовах;

Вторинна база даних. WikiData може записувати не тільки твердження, але і їх джерела, що відображає різноманітність наявних знань і їх надійність.

Збір структурованих даних. Це дозволить легко повторно використовувати ці дані і дозволить комп'ютерам легко обробляти їх.

Підтримка проектів Вікімедіа. WikiData підтримує Вікіпедію з більш зручними посиланнями на різні мови та інфобоксами, таким чином, зменшуючи навантаження на Вікіпедію та підвищуючи її якість.

WikiData API знаходиться у активній розробці, але є бібліотека Wikidata Toolkit на Java, що дозволяє отримати клієнтський доступ до WikiData.

1.4. Висновок

Нижче наведена таблиця (табл. 2), в якій показані збірні дані по розглянутим базам. З точки зору програміста-розробника найзручнішою є FreeBase – має доступний API від Google, є доступ до інфобоксів, досить обширна, MQL зручніша за інші аналоги, але проект закритий, дані тільки у режимі читання, вся інформація переноситься на сервери WikiData. В свою чергу WikiData працює, але знаходиться у стані активної розробки, тому використання API може бути з помилками. Залишається DBpedia – база знань, на якій буде побудована робота плагіна. Я зробив свій вибір на користь DBpedia тому що ця база є основою для всього LOD, постійно розширюється, стабільно працює та має кінцеві точки для запитів на SPARQL. Програмний продукт дипломної роботи буде побудовано на основі інтерпретації запитів за допомогою фрейворку query на мову SPARQL та отриманні даних з DBpedia.

Таблиця 2 – Порівняння баз знань

База	API	Мова запитів	Стан проекту	Інтерфейс	Ліцензія	Особливості
DBpedia	-	SPARQL	підтримується	Відсутній, тільки по запиту SPARQL	GNU Free Documentation License	Найбільша база знань. Ядро LOD
FreeBase	+	SQL	на етапі закриття	Дружній до користувача	Creative Commons Attribution	Має зручні інфобокси з Вікіпедії, та доступ до них
WikiData	- /Java lib	WDQ	в активній розробці	Повністю повторює Вікіпедію, багато зайвої інформації	Creative Commons Public Domain Dedication	Багатомовна(більш за інші), найбільш перспективна

2. ПРАКТИЧНА ЧАСТИНА

2.1. Робота з базою знань DBpedia

Для того, щоб показати специфіку роботи зі SPARQL та його відмінності від звичайних мов запитів типу SQL, буде продемонстровано кілька прикладів. Візьмемо приклад одного запиту і послідовно покажемо, як правильно побудувати його SPARQL-аналог.


Задача:

Вивести список зі 100 компаній, які були засновані молодими діячами культури і мистецтва. Вивести фотографію засновника і його короткий опис.

Кожен SPARQL-запит надсилається не в невизначену невідому точку Всесвітньої павутини, а навпаки, задається по відношенню до строго визначеної множини RDF-даних. В даному прикладі запити будемо посилати на DBpedia - одного з найбільших центрів інформації в Linked Data, розглянутого у попередньому розділі. Оскільки, DBpedia - це проект створення структурованої і постійно оновлюваної копії Вікіпедії, представленої в стандарті RDF, то ми можемо розраховувати на наступну структуровану інформацію: Категорії, Перенаправлення і найголовніше, Інформаційні бокси. Більша частина даних DBpedia - це переведені в RDF інформаційні бокси, які розташовуються на багатьох сторінках Вікіпедії.

Автоматичні скрипти регулярно сканують Вікіпедію і намагаються грамотно перетворити текстовий вміст Інформаційних боксів в структуровані RDF-дані - дати, числа, рядки та унікальні ідентифікатори (URI).

Розглянемо, наприклад, сторінку, присвячену Леонардо да Вінчі (Рис. 8). Праворуч від тексту знаходиться той самий інформаційний бокс, який буде перетворений в RDF-трійки.



dbpedia:Leonardo_da_Vinci	dbpprop:name	Leonardo da Vinci@en.
dbpedia:Leonardo_da_Vinci	dbpprop:caption	Self-portrait in red chalk, circa 1512 ...of Turin@en .
dbpedia:Leonardo_da_Vinci	dbpprop:birthName	Leonardo di ser Piero da Vinci@en.
dbpedia:Leonardo_da_Vinci	dbpprop:placeOfBirth	Anchiano by Vinci, Italy@en .
dbpedia:Leonardo_da_Vinci	dbpprop:dateOfBirth	1452-04-15^^xsd:date.
dbpedia:Leonardo_da_Vinci	dbpedia-owl:thumbnail	<http://upload.wi.../200px-Leonardo_self.jpg>.
dbpedia:Leonardo_da_Vinci	dbpprop:nationality	dbpedia:Italy .
dbpedia:Leonardo_da_Vinci	dbpprop:field	Many and diverse fields of arts and sciences@en .
dbpedia:Leonardo_da_Vinci	dbpedia-owl:field	dbpedia:The_arts .
dbpedia:Leonardo_da_Vinci	dbpedia-owl:movement	dbpedia:High_Renaissance .
dbpedia:Leonardo_da_Vinci	dbpprop:works	Mona Lisa, The Last Supper, The Vitruvian

Рис. 8 – Інфобокс з Вікіпедії (ліворуч) та сторінка Лонардо Да Вінчі у DBpedia (праворуч)

Графічно ці дані можуть бути відображені наступним чином (Рис. 9):

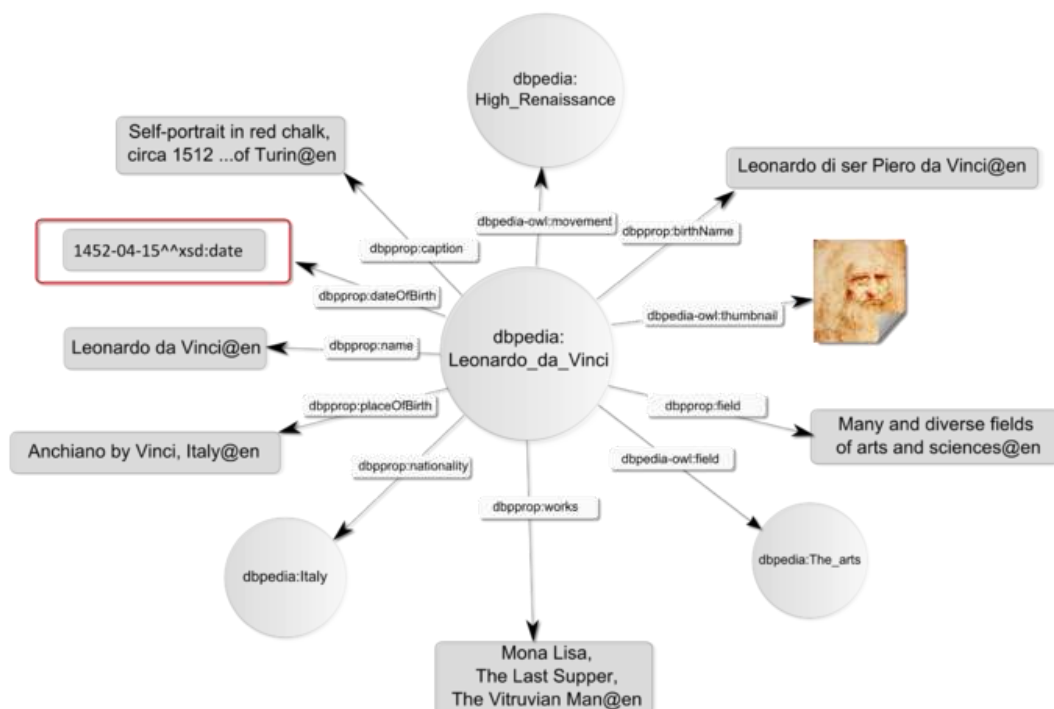


Рис. 9 – Графічне представлення структурованих даних (кружки – ідентифікатори ресурсів, квадрати - літерали) [7]

Спочатку спробуємо просто вивести сто компаній, використовуючи тип запиту SELECT та отримаємо значення змінних. Точка доступу заповнить змінну всіма суб'єктами, від яких йдуть зазначені нами властивості до зазначених нами значень.

Для обмеження кількості виведених результатів знадобиться ключове слово LIMIT.

```
SELECT * WHERE
{
  ?a <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://dbpedia.org/ontology/Company> .
}
LIMIT 100
```

Зірочка після SELECT означає, що треба побачити значення всіх змінних, згаданих у запиті. У даному запиті у нас всього лише одна змінна ?a. У SPARQL всі змінні починаються з знаку питання. Як бачите, запит укладений у фігурні дужки, а обмеження на виведення результату (модифікатор LIMIT) стоїть після нього.

У цьому запиті ми використовуємо повний URL класу Компанія: <http://dbpedia.org/ontology/Company> і повний URL властивості Тип з мови RDF: <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>. Кожен раз, коли використовуємо в запиті URL, необхідно брати його в трикутні дужки. Таким чином, наш запит можна інтерпретувати так: "вивести 100 суб'єктів, що мають властивість Тип зі значенням Компанія." Зверніть увагу, що порядок дотримання важливий: не можна просто переставити місцями змінні і URL'и.

У RDF всі об'єкти, властивості і класи позначаються своїми ідентифікаторами URL. Однак, якщо завжди використовувати повний URL-запис в SPARQL-запитах, їх буде складніше читати і розуміти.

На щастя в SPARQL, як і в RDF є механізм скорочення URL. На початку запиту можна задати аббревіатури для ресурсів. Такі аббревіатури називаються

префіксами і оголошуються за допомогою конструкції PREFIX ім'я_префікса:
<скорочуваний URL>:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

SELECT * WHERE
{
  ?a rdf:type dbpedia-owl:Company .
}
LIMIT 100
```

Нас цікавлять лише компанії, засновані представниками культури і мистецтва, імена цих засновників, їхні фотографії.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>
SELECT * WHERE
{
  ?companyURL rdf:type dbpedia-owl:Company;
    dbpprop:companyName ?corporation ;
    dbpedia-owl:foundedBy ?founderURL .
  ?founderURL rdf:type dbpedia-owl:Artist .
}
LIMIT 100
```

Далі додаємо фільтрацію за датою народження *FILTER* (*?founderBirth > "1940-03-10"^^xsd:date*), опціональні поля з фотографією за допомогою *OPTIONAL* { *?founderURL foaf:depiction ?founderPicture .* }, видалимо дублювання з ключовим словом *DISTINCT*. В кінцевому вигляді повний запит виглядатиме так:


```

SELECT DISTINCT ?corporation ?founderName ?founderPicture ?founderBirth
?founderDescription WHERE
{
  ?companyURL rdf:type          dbpedia-owl:Company ;
              dbpprop:companyName ?corporation      ;
              dbpedia-owl:foundedBy ?founderURL      .
  ?founderURL dbpedia-owl:birthDate ?founderBirth    ;
              foaf:name           ?founderName      ;
              dbpedia-owl:abstract ?founderDescription .
  OPTIONAL
  {
    ?founderURL foaf:depiction      ?founderPicture .
  }
  FILTER (?founderBirth > "1940-03-10"^^xsd:date ) .
  FILTER langMatches( lang(?corporation), "EN" ) .
  FILTER langMatches( lang(?founderDescription), "EN" ) .
  { ?founderURL rdf:type  dbpedia-owl:Artist. }
  UNION
  { ?founderURL rdf:type  dbpedia-owl:Actor. }
  UNION
  { ?founderURL rdf:type  dbpedia-owl:MartialArtist. }
  UNION
  { ?founderURL dbpedia-owl:occupation dbpedia:Martial_arts.}
}
LIMIT 100

```

Результатом запиту у вигляді скріншоту (Рис. 10). Метою цього прикладу було показати гнучкість та універсальність мови SPARQL. Широкий спектр можливостей дозволяє точно отримувати відповідь на поставлене питання, тому в наступному підрозділі буде детально розглянуто модель розробленого семантичного додатку до браузеру та використані підходи та алгоритми у обробці людської мови для генерування запитів SPARQL [5].

corporation	founderName	founderPicture	founderBirth	founderDescription
"Basement Comics"@en	Budd Root"@en		"1938-02-00"^^<http://www.w3.org/2001/XMLSchema#Year>	Budd Root (b. 1938 in Heidelberg, Germany) is an American cartoonist, and creator of the independent comic book Cavevoman, published by Root's company Basement Comics/Amryl Entertainment. Root is one of the premiere contemporary "good girl" artists, with an art style similar to Art Adams. @en
"Basement Comics"@en	Root Budd"@en		"1938-02-00"^^<http://www.w3.org/2001/XMLSchema#Year>	Budd Root (b. 1938 in Heidelberg, Germany) is an American cartoonist, and creator of the independent comic book Cavevoman, published by Root's company Basement Comics/Amryl Entertainment. Root is one of the premiere contemporary "good girl" artists, with an art style similar to Art Adams. @en
"Eto Books"@en	Ki Longfellow"@en	http://en.wikipedia.org/wiki/Special:FilePath/Ki_Nymph_cropping	1944-12-09-02:00	Ki Longfellow (born Pamela Kelly) is an American novelist, playwright, theatrical producer, theater director and entrepreneur. She is best known in the United States for her novel <i>The Secret Magdalene</i> (2005). This is among her recent works exploring the divine feminine. In England, she is likely best known as the widow of Vivian Stanshall, the late musician, singer-songwriter, author, broadcaster, and wit. Longfellow started writing seriously after Stanshall's death. Her first two novels, <i>China Blues</i> (1989) and <i>Chasing Women</i> (1992) and the recent <i>Houdini Heart</i> (2011) are mysteries and thrillers. In April 2013, the first three titles of her <i>Sam Russo Mysteries</i> were published, part of a four series set in and around New York City in the late 1940s. <i>Walks Away Woman</i> —about a neglected Arizona housewife walking out into the Sonoran Desert to die, was completed in 2002 but not published until December 2013. @en
"Eto Books"@en	Longfellow, Ki"@en	http://en.wikipedia.org/wiki/Special:FilePath/Ki_Nymph_cropping	1944-12-09-02:00	Ki Longfellow (born Pamela Kelly) is an American novelist, playwright, theatrical producer, theater director and entrepreneur. She is best known in the United States for her novel <i>The Secret Magdalene</i> (2005). This is among her recent works exploring the divine feminine. In England, she is likely best known as the widow of Vivian Stanshall, the late musician, singer-songwriter, author, broadcaster, and wit. Longfellow started writing seriously after Stanshall's death. Her first two novels, <i>China Blues</i> (1989) and <i>Chasing Women</i> (1992) and the recent <i>Houdini Heart</i> (2011) are mysteries and thrillers. In April 2013, the first three titles of her <i>Sam Russo Mysteries</i> were published, part of a four series set in and around New York City in the late 1940s. <i>Walks Away Woman</i> —about a neglected Arizona housewife walking out into the Sonoran Desert to die, was completed in 2002 but not published until December 2013. @en
"Mam Tor Publishing"@en	Liam Roger Sharp"@en	http://commons.wikimedia.org/wiki/Special:FilePath/Mr_Ughr_Mae_02.jpg	1968-05-02-02:00	Liam Roger Sharp (born 2 May 1968) is a British comic book artist, writer and publisher. @en
"Mam Tor Publishing"@en	Sharp, Liam"@en	http://commons.wikimedia.org/wiki/Special:FilePath/Mr_Ughr_Mae_02.jpg	1968-05-02-02:00	Liam Roger Sharp (born 2 May 1968) is a British comic book artist, writer and publisher. @en
"HandMade Films"@en	George Harrison"@en	http://commons.wikimedia.org/wiki/Special:FilePath/George_Harrison_1974_sitting.jpg	1943-02-25-02:00	George Harrison, MBE (25 February 1943 – 29 November 2001) was an English musician, singer and songwriter who achieved international fame as the lead guitarist of the Beatles. Although John Lennon and Paul McCartney were the band's primary songwriters, most of their albums included at least one Harrison composition, including "White My Guit", "Gently Weeps", "Here Comes the Sun" and "Something", which became the Beatles' second-most-covered song. Harrison's earliest musical influences included Big Bill Broonzy, George Formby and Django Reinhardt. Chet Atkins, Chuck Berry and Ry Cooder were significant later influences. By 1965 he had begun to lead the Beatles into folk rock through his interest in the Byrds and Bob Dylan, and towards Indian classical music through his use of the sitar on "Norwegian Wood (This Bird Has Flown)". He developed an interest in the Hare Krishna movement and became an admirer of Indian culture and mysticism, introducing them to the other members of the Beatles and their Western audience by incorporating Indian instrumentation in their music. After the band's break-up in 1970, Harrison released the triple album <i>All Things Must Pass</i> , from which two hit singles originated. He also organized the 1971 Concert for Bangladesh with Navi Shabar, a precursor for later benefit concerts such as Live Aid. Harrison was a music and film producer as well as a musician; he founded Dark Horse Records in 1974 and co-founded HandMade Films in 1978. Harrison released several best-selling singles and albums as a solo performer, and in 1988 co-founded the platinum-selling supergroup the Traveling Wilburys. A prolific recording artist, he was featured as a guest guitarist on tracks by Badfinger, Ronnie Wood and Billy Preston, and collaborated on songs and music with Bob Dylan, Eric Clapton and Tom Petty, among others. Rolling Stone magazine ranked him number 11 in their list of the "100 Greatest Guitarists of All Time". Harrison's first marriage, to Pattie Boyd, ended in divorce in 1977. The following year he married Olivia Trinidad Arias, with whom he had one son, Dhani. Harrison died in 2001, aged 58, from lung cancer. He was cremated and his ashes were scattered in the Ganges and Yamuna rivers in India, in a private ceremony according to Hindu tradition. He left almost £100 million in his will. @en
"HandMade Films"@en	George Harrison"@en	http://commons.wikimedia.org/wiki/Special:FilePath/George_Harrison_1974_sitting.jpg	1943-02-25-02:00	George Harrison, MBE (25 February 1943 – 29 November 2001) was an English musician, singer and songwriter who achieved international fame as the lead guitarist of the Beatles. Although John Lennon and Paul McCartney were the band's primary songwriters, most of their albums included at least one Harrison composition, including "White My Guit", "Gently Weeps", "Here Comes the Sun" and "Something", which became the Beatles' second-most-covered song. Harrison's earliest musical influences included Big Bill Broonzy, George Formby and Django Reinhardt. Chet Atkins, Chuck Berry and Ry Cooder were significant later influences. By 1965 he had begun to lead the Beatles into folk rock through his interest in the Byrds and Bob Dylan, and towards Indian classical music through his use of the sitar on "Norwegian Wood (This Bird Has Flown)". He developed an interest in the Hare Krishna movement and became an admirer of Indian culture and mysticism, introducing them to the other members of the Beatles and their Western audience by incorporating Indian instrumentation in their music. After the band's break-up in 1970, Harrison released the triple album <i>All Things Must Pass</i> , from which two hit singles originated. He also organized the 1971 Concert for Bangladesh with Navi Shabar, a precursor for later benefit concerts such as Live Aid. Harrison was a music and film producer as well as a musician; he founded Dark Horse Records in 1974 and co-founded HandMade Films in 1978. Harrison released several best-selling singles and albums as a solo performer, and in 1988 co-founded the platinum-selling supergroup the Traveling Wilburys. A prolific recording artist, he was featured as a guest guitarist on tracks by Badfinger, Ronnie Wood and Billy Preston, and collaborated on songs and music with Bob Dylan, Eric Clapton and Tom Petty, among others. Rolling Stone magazine ranked him number 11 in their list of the "100 Greatest Guitarists of All Time". Harrison's first marriage, to Pattie Boyd, ended in divorce in 1977. The following year he married Olivia Trinidad Arias, with whom he had one son, Dhani. Harrison died in 2001, aged 58, from lung cancer. He was cremated and his ashes were scattered in the Ganges and Yamuna rivers in India, in a private ceremony according to Hindu tradition. He left almost £100 million in his will. @en

Рис. 10 – Результат запису

2.2. Концепція семантичного плагіна Semantic Quick Search (SQS)

Об'єм доступної інформації в інтернеті зростає з кожним днем, існує потреба в структуризації та смислового зв'язуванні розрізнених даних, так з'явилась концепція семантичного веб. Наступний крок – це швидкий та доступний пошук по базі знань. На даний момент семантичний веб включає в себе достатні ресурси для існування великої бази знань (Open Linked Data), яка обмежена двома факторами: семантичний веб містить лише ті зв'язки, які може собі уявити людина, котра їх створює, другий фактор – велика кількість нерозміченої інформації. Проте існує ще й проблема відсутності дружнього до користувача інтерфейсу. Немає можливості візуалізувати запити до бази знань у зручній для користувача формі, результати отримуються у важкому для сприйняття вигляді.

Для вирішення поставленої задачі був розроблений проект розширення до браузеру Google Chrome під назвою Semantic Quick Search (SQS). Плагін або розширення – це програмний модуль, який можна вбудувати в готову систему (в цьому випадку – браузер Google Chrome) та розширити її можливості.

Уявімо ситуацію, коли користувач, читаючи текстову інформацію в інтернеті зустрічає незнайоме слово або термін. Зазвичай, це слово виділяється, копіюється та шукється у Google у новій вкладці. Далі користувачу треба результати пошуку перебрати та знайти те, що його цікавить (при умові, що пошуковий запит був коректний). Вочевидь ці маніпуляції по пошуку потрібної інформації потребують час, а все задля того, щоб, в ідеалі, знайти сторінку в Вікіпедії та прочитати інфобокс. Для отримання бажаного результату користувачеві потребувалось зробити мінімум 7 кліків мишкою або 5 кліків та 2 використання гарячих клавіш. Як бачимо ця ситуація може бути чудово вирішена за допомогою запиту до DBpedia, адже в ній знаходяться структуровані дані з Вікіпедії, зокрема інфобокси.

Серед обмалі робочих програмних продуктів по роботі із запитами до семантичних даних у Веб можна виділити Knowledge Graph від Google (Рис. 11), який видає точний, але обмежений результат одночасно зі звичайним запитом до пошукової системи. Це можна вважати частковим рішенням проблеми.

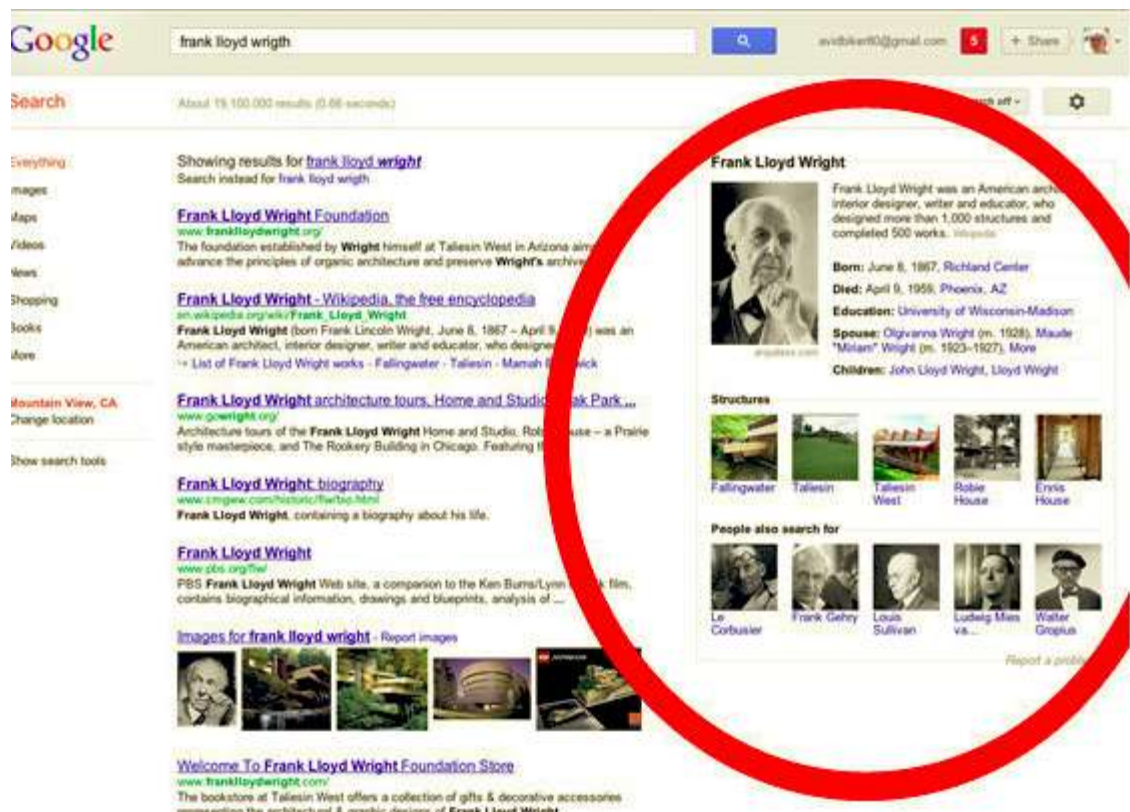


Рис. 11 - Knowledge Graph від Google

Отже, **головна ідея плагіна** заключається в наступному: користувач виділяє незнайоме слово та у контекстному меню правої клавіші мишки знайти строку DBpedia Search або Freebase Search та отримати результат у випадаючому вікні. Або натиснути на іконку SQS у верхньому кутку(Рис. 12), написати питання у строку пошуку та отримати відповідь там же. З точки зору розробника – основні алгоритми роботи розширення направлені на аналіз природньої мови, парсинг тексту та генерація запиту SPARQL або MQL до відповідної бази знань. Також приділяється увага зручності у користуванні,

мінімалізації кількості налаштувань та кількості кліків для досягнення користувачем мети пошуку.



Рис.12 – Семантичний плагін встановлений у Google Chrome

Концепція розширення являє собою рішення поставленої задачі по візуалізації запитів до ресурсів LOD, а також пропонує реальне, зручне рішення проблеми відсутності графічного інтерфейсу користувача для семантичних додатків.

2.3. Розробка розширення. Доцільність проекту

2.3.1. Вимоги та технічне завдання до розширення Semantic Quick Search (SQS)

1. Призначення

ПЗ призначене для перетворення ключового слова у пошуковий запит на мові SPARQL та MQL до семантичних баз знань DBpedia та Freebase відповідно, виведення результату запиту у ергономічне випадаюче вікно на веб сторінці.

2. Вимоги до ПЗ

2.1. Вимоги до функціональних характеристик

2.1.1. Програма має забезпечувати можливість виконання наступних функцій:

- Пошук виділеного слова або словосполучення по вибраній базі знань
- Трансформація ключового слова у SPARQL запит
- Можливість вибору бази знань: DBpedia та Freebase
- Вивід результату запиту у випадаюче вікно
- Пошук по ключовому слову за 3 кліка
- Можливість задати питання природньою мовою у рорир меню розширення

2.1.2. Вхідні дані:

- Виділене ключове слово чи словосполучення
- Питання природньою мовою

2.1.3. Організація вхідних та вихідних даних

Вхідні дані виділяються курсором або вводяться у вигляді питання у рорир меню. Вихідні дані виводяться або у випадаюче вікно на веб сторінці, або у рорир меню розширення

2.2. Вимоги до надійності програми

Передбачення контролю вхідної інформації , а також заборона дій користувача, які призведуть до неправильного функціонування програми.

При виявленні помилки, на екран буде виведено відповідне повідомлення.

2.3. Вимоги до складу та параметрів технічних засобів

Аналогічна вимогам браузеру Google Chrome

2.4. Вимоги до програмної сумісності

Розширення розроблене для браузеру Google Chrome.

3. Вимоги до програмної документації

Програмною документацією слугує звітно-пояснювальна записка та автоматично генерована документація програмного коду.

2.3.2. Засоби розробки SQS

Компанія Google у своєму браузері Chrome передбачила можливість встановлювати сторонні додатки у вигляді розширень, що дозволяють користувачу додати бажані можливості до роботи браузеру. Оскільки розширення не публікується на маркеті Chrome, то слід включити режим розробника у браузері, щоб з'явилась можливість встановлювати власні програми.

Типове розширення для Chrome складається з файлу маніфесту і деякої комбінації з фонові сторінки, сторінок користувача інтерфейсу і сценаріїв.

Все починається з файлу маніфесту з ім'ям manifest.json. Цей файл містить інформацію, необхідну Chrome для того, щоб завантажити розширення - заголовок, опис, необхідні дозволи, ярлик і т.п. Приклад:

```
{
  "manifest_version": 2,
  "name": "Semantic Quick Search",
  "version": "1.0",
  "description": "Extension for a quick semantic search trough
  DBpedia/Freebase",
```

```
"browser_action": {  
  "default_title": "SQS!",  
  "default_icon": "img/icon.png",  
  "default_popup": "popup.html"  
}, ...
```

Фонова сторінка - це сторінка, яка запускається і виконується в своєму власному контексті, незалежно від того, скільки вкладок або вікон відкрито. У розширення може бути тільки одна фонові сторінка. Вона потрібна для розширень, де повинен виконуватися процес, що охоплює всі сторінки Chrome.

Сторінки графічного інтерфейсу (UI Pages) - це будь-які сторінки, які розширення представляє користувачеві. Це може бути спливаюче вікно, сторінка властивостей, сторінка в складі розширення або сторінка, що замінює сторінку Chrome за замовчуванням (наприклад, сторінка нової вкладки).

Сценарії - це файли JavaScript, які впроваджуються в Web-сторінки для взаємодії з ними, нижче приведений приклад створення строчки контекстного меню. Сценарії виконуються у власному ізольованому контексті, але можуть отримувати доступ до моделі DOM сторінки. Сценарії можуть встановлювати зв'язок з іншими сторінками розширення за допомогою спеціального API передачі повідомлень.


```
// Set up context menu tree at install time.
chrome.runtime.onInstalled.addListener(function () {
    var contexts = ["page", "selection", "link", "editable", "image",
"video",
                    "audio"];
    for (var i = 0; i < contexts.length; i++) {
        var context = contexts[i];
        var title = "Search with SQS";
        var id = chrome.contextMenus.create({
            "title": title, "contexts": [context],
            "id": "context" + context
        });
        console.log("'" + context + "' item:" + id);
    }
});
```

Microsoft Visual Studio була вибрана як IDE для розробки (там теж присутні засоби сборки Java Script коду), був створений шаблон для розробки саме додатку для Chrome. Visual Studio була вибрана виходячи з того, що в ній інтегрована система IntelliSense – технологія автодоповнення коду, – а також система динамічної перевірки коду на помилки без сборки всього проекту. Рішення від Microsoft виявилось найзручнішим.

В проекті присутня частина написана на Python, що розроблялась на Ubuntu. Створений локальний сервер, що хостить сторінку на Python з бібліотекою query для аналізу природньої мови та перетворення її на запит SPARQL.

2.3.3. Доцільність концепції

Розширення для браузера може знадобитися з кількох причин. Зазвичай розширення використовуються для зв'язку браузера з іншим додатком або службою. Цю задачу вирішують Evernote, 1Password і Adobe Shadow і безліч інших розширень. Або може знадобитися додати в браузер якісь нові, відсутні функції, інструменти розробки або засоби захоплення зображень з екрану. Існують вузькоспеціалізовані розширення - для відстеження спортивних результатів, удосконалення конкретних Web-сайтів, спостереження за прогнозами погоди і т.п. Розширення використовуються для самих різних цілей.

Серед концепцій, що були розглянуті, також був Java аплет, веб сторінка з пошуком по базі знань та ПЗ для Windows, що оброблює дамп бази знань, збережений локально на комп'ютері. Java аплети, через свою незахищеність стали блокуватися системою захисту браузерів, тому варіант не підійшов. Наступним кандидатом була сторінка з пошуком, але такий підхід не був оригінальним – сторінок пошукачів вже існує достатньо, також користувачу доводилось би витратити більше часу для пошуку потрібної інформації (задача була розробити зручний інтерфейс для візуалізації запитів). Остання концепція не мала сенсу, адже могла використовуватися лише у дослідницьких цілях (встановлення дампу на ПК та його регулярне оновлення не можуть претендувати на зручний інтерфейс).

Саме тому було вирішено зупинитися на концепції розширення для браузера. Розширення надають досить широкий спектр можливостей, можуть бути легко портовані на інші браузери, дозволяють зробити пошук дійсно зручним.

Також для були знайдені зручні засоби аналізу природньої мови `query` для Python.

2.4. Аналіз тексту. Алгоритми NLP

Не секрет, що текстові інтерфейси відіграють важливу роль тому що вони найбільш інтуїтивні у використанні для кінцевого користувача. Вони підтримуються як на ПК так і в веб додатках та легко доступні на всіх видах девайсів.

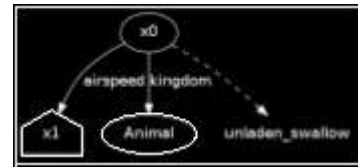
Більшість баз знань надають можливість для запитів за допомогою формальних мова, наприклад, SPARQL або MQL. Але вони мають досить складний синтаксис, вимагають хорошого розуміння онтології даних та мають велику вірогідність помилки, через необхідність набирати довгі і складні URI. Ці мови аналогічні принципам SQL для запитів до традиційних реляційних баз даних і не повинні розглядатися в якості інструменту для кінцевого користувача.

Різні методи зручного доступу знань були розроблені раніше. Деякі з них надають графічний інтерфейс, де користувачі можуть переглядати онтологію, інші пропонують форми-інтерфейси для семантичного пошуку, базовані на онтології, приховуючи складність формальних мов. Найбільш складним вважається підхід, який надає простий текстовий бокс для запиту, що оброблює природню мову (NL) та перетворює її на формальний запит.

У проекті SQS використовуються алгоритми фреймворку query, які базуються на парсингу вхідних даних. Алгоритм покладений в основу цього фреймворку працює на основі NLTK – natural language toolkit, провідна платформа для розробки питоновських програм по роботі та аналізу природньої мови, платформа надає можливості для класифікації, розбиття речень, розмічування, парсингу та семантичного обґрунтування.

Розглянемо поетапно алгоритм query для речення *What is the airspeed velocity of an unladen swallow?*

1. Парсинг речення: *What/what/WP is/be/VBZ the/the/DT airspeed/airspeed/NN velocity/velocity/NN of/of/IN an/an/DT unladen/unladen/JJ swallow/swallow/NN*



2. Пошук збігу + проміжне представлення
3. Генерування запиту та DSL (Domain Specific Languages):

```

SELECT DISTINCT ?x1 WHERE {
  ?x0 kingdom "Animal".
  ?x0 name "unladen swallow".
  ?x0 airspeed ?x1.
}
  
```

Парсинг речення відбувається на рівні слів, а не символів, по заданій схемі:

Word = Token + Lemma + POS (part of speech tagset)

Далі NLTK має достатньо можливостей для подальшої обробки речення.

Всі види питань задаються регулярними виразами:

Token("what") + Lemma("be") + Question(Pos("DT")) + Plus(Pos("NN"))

Проміжне представлення являє собою граф, в якому деякі значення відомі, а деякі ні x_0 , x_1 . Граф аналогічний представленню даних в базах знань.

Далі йде процес генерації запиту, відбувається на основі регулярних виразів. Кожне окреме питання потрібно задавати також у вигляді регулярного виразу, як показано вище [6,7].

2.5 Архітектура SQS

Архітектура плагіна являє собою два окремі програмні модулі: клієнтська частина (розширення до браузеру) та серверна (парсер, обробник запиту, query). Запити формуються на основі багатого вибору регулярних виразів для питань та речень на природній мові, для парсингу запиту в query використовується POS NLTK tagger - parts of speech natural language toolkit – потужний фреймворк для Python, що класифікую звичайні слова за частинами мови: іменники, дієслова, прислівники та прикметники (табл. 3). Це надзвичайно важливо для побудови регулярних виразів для аналізу речень на людській мові. Обов’язково слід згадати, що проект працює виключно з англійською мовою, це обумовлено тим, що словарний корпус NLTK наявний тільки на англійській мові.

Таблиця 3 – Абревіатури тегів частин мови

ADJ	adjective	<i>new, good, high, special, big, local</i>
ADP	adposition	<i>on, of, at, with, by, into, under</i>
ADV	adverb	<i>really, already, still, early, now</i>
CONJ	conjunction	<i>and, or, but, if, while, although</i>
DET	determiner, article	<i>the, a, some, most, every, no, which</i>
NOUN	noun	<i>year, home, costs, time, Africa</i>
NUM	numeral	<i>twenty-four, fourth, 1991, 14:24</i>
PRT	particle	<i>at, on, out, over per, that, up, with</i>
PRON	pronoun	<i>he, their, her, its, my, I, us</i>
VERB	verb	<i>is, say, told, given, playing, would</i>
.	punctuation marks	<i>., ; !</i>
X	other	

Приклад NLTK регулярного виразу:

```
regex = Question(Pos("DT")) + Plus(Pos("NN") | Pos("NNP"))
```

Повертаючись до розгляду архітектури слід розглянути UML діаграму класів (рис. 13). Наглядно показана клієнт-серверна структура, що побудована на протоколі GET HTTP (localhost:8000/cgi-bin/server.py?base_btn=dbpedia&Ask_box=nauru).

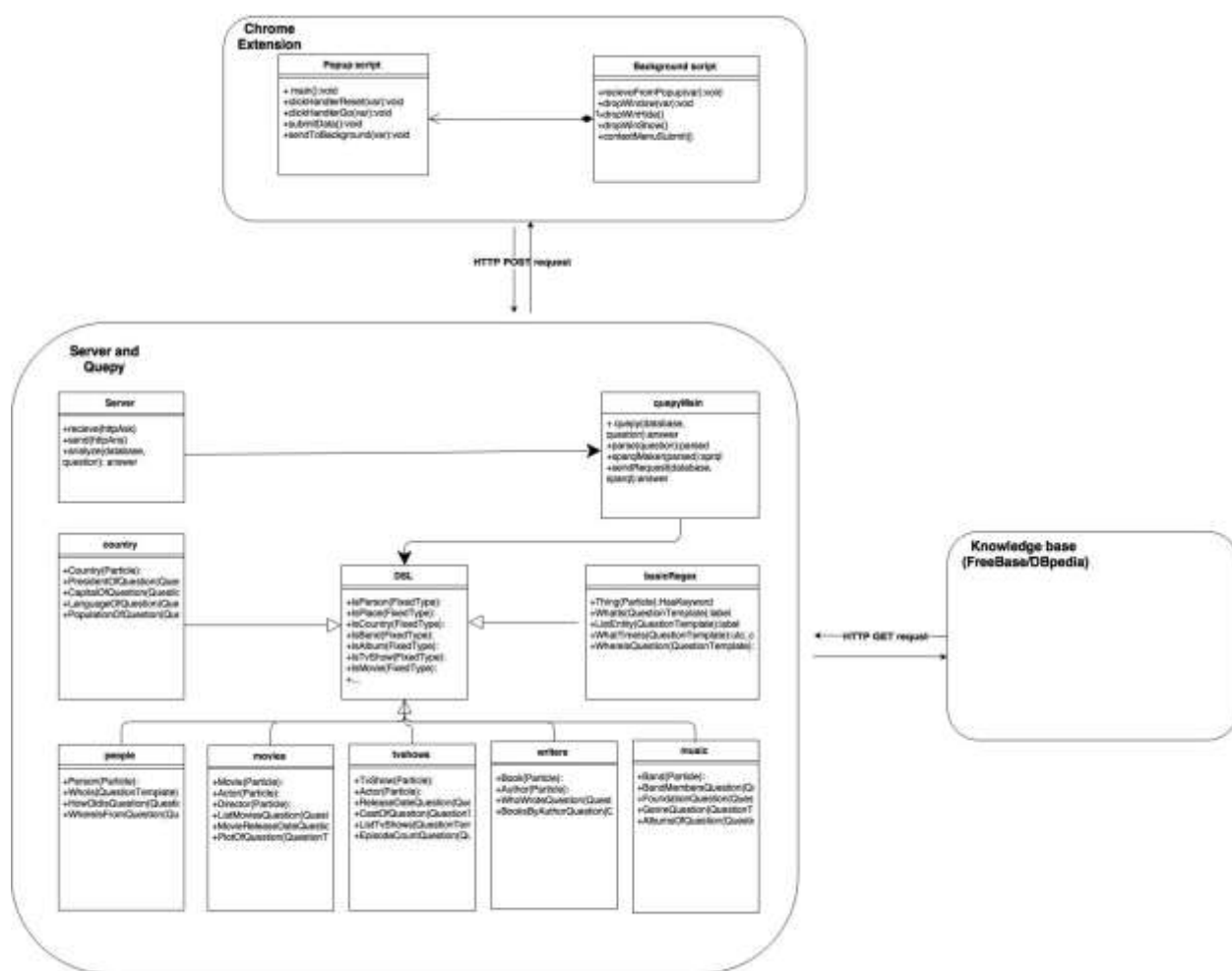


Рис.13- UML діаграма класів та потоки даних SQS

Клієнтська частина містить 2 класи Java Script: popup та background. Перший описує послідовність дій та реалізує функції для випадального вікна

розширення (вибір бази знань та відправка запиту на сервер), зовнішній вигляд описується звичайною html сторінкою. Скрипт обробнику події натискання кнопки Go або Clear, доступ до елемента «кнопка» здійснюється через пошук 'button' в об'єкті батьківського документу.

```
document.addEventListener('DOMContentLoaded', function () {
    document.querySelectorAll('button')[0].addEventListener('click',
clickHandlerGo);
    document.querySelectorAll('button')[1].addEventListener('click',
clickHandlerReset);
});
```

Серверна частина складається з модулю прийому-передачі даних на клієнт, аналізаторі, побудованому на query, та модулі, що посилає запит на базу знань. Оскільки query – це фреймворк для створення ПЗ, то його базова структура обов'язково включає клас опису регулярних виразів речень та Domain Specific Language – клас, що задає описи виразів з RDF Schema специфічні для кожної онтології: назви відношень та ін.

```
class IsPerson(FixedType):
    fixedtype = "foaf:Person"
class IsPlace(FixedType):
    fixedtype = "dbpedia:Place"
```

Можливості SQS плагіна включають обробку широкого спектру природніх запитів на теми:

- Географія - країни, президенти, мови, столиці, населення;
- Фільми - фільми, актори, режисери, тривалість, реліз, сюжет;
- Музика - групи, члени груп, дата заснування, жанр, альбоми;
- Люди – персони, опис, вік, походження;
- Шоу – актори, шоу, реліз, кількість епізодів, режисери;
- Письменники – книги, автори, опис.

Кожен запит можна задати у вигляді питання (What Is, List smth, What Time Is, Where Is) або просто по ключовому слову.

UML діаграма показує тільки статичну інформацію про стан та структуру проекту, тому для повної картини слід розглянути діаграму послідовностей (рис. 14). SQS має доволі очевидну єдину функцію – швидкий пошук, тож нижче показана послідовність для процесу пошуку інформації:

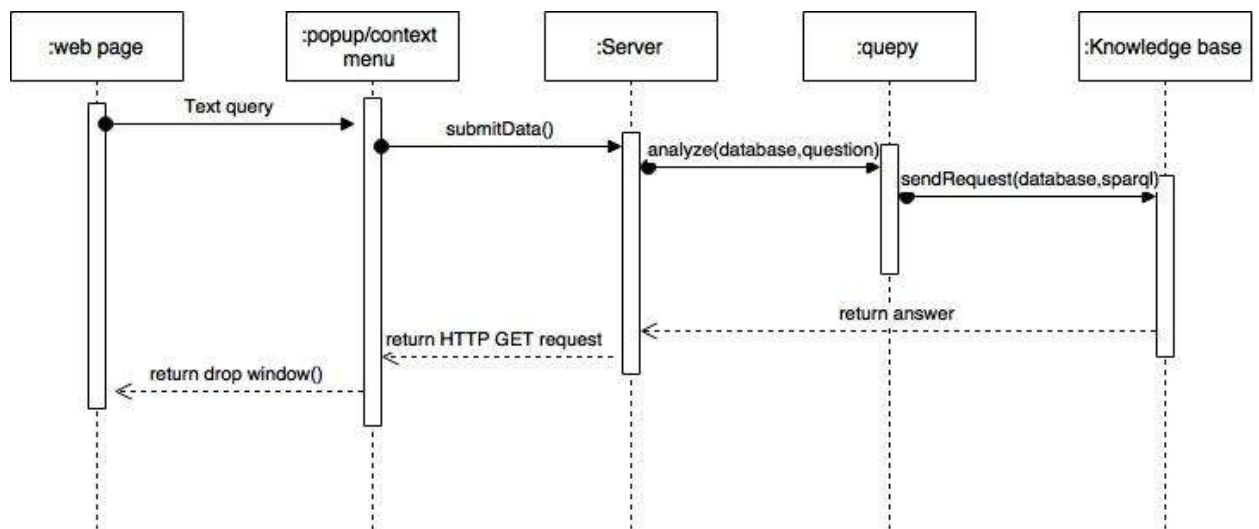


Рис.14 – Діаграма послідовності

Користувач посилає текст або з сторінки браузера, або через рорир віконце, спрацьовує метод `submitData()`, що посилає HTTP GET request на сервер, дані аналізуються `query` та посилається запит на базу знань. Результат у вигляді html коду відправляється назад на клієнт, де вбудовується на сторінку у вигляді вспливаючого вікна на затемненому фоні.



Рис. 15 – результат запиту.

2.6 Висновок

В якості висновку слід сказати, що запропонована система має великий потенціал, адже має мінімалістичний інтерфейс, просте управління, мінімум налаштувань, та, як наслідок, максимум результату. Архітектура клієнт-сервер дозволяє перекласти відповідні комп'ютерні розрахунки на сервер, що дозволяє не загрузати клієнтську частину. Це дуже важливо, якщо врахувати, що клієнтська частина реалізується у вигляді розширення до браузера – будь-які затримки в роботі будуть затримувати роботу всього браузера.

3. РЕАЛІЗАЦІЯ ТА ПРИКЛАДИ РОБОТИ

3.1 Реалізація

Створення програмного продукту проходило в строго запланованій послідовності етапів проектування (каскадна модель):

- Бізнес моделювання – формування основи логіки предметної області
- Створення та аналіз вимог – ТЗ
- Планування
- Розробка архітектури – діаграма класів, пакетів, data flow та use case
- Кодування – реалізація архітектури
- Тестування – перевірка на відповідність вимогам
- Створення документації – оформлення звіту

Важливим питанням на етапі кодування було вивчення обмежень безпеки Google для створення сторонніх розширень браузеру або CSP (content security Policy). Основні обмеження припадають на деякі java script функції, що можуть призупиняти роботу браузеру (setTimeout()), функцію eval() – може бути використана для XSS атаки. Як неправильно:

```
alert(eval("foo.bar.baz"));  
window.setTimeout("alert('hi')", 10);
```

Як правильно:

```
alert(foo && foo.bar && foo.bar.baz);  
window.setTimeout(function() { alert('hi'); }, 10);  
window.setInterval(function() { alert('hi'); }, 10);  
function() { return foo && foo.bar && foo.bar.baz };
```

Також в описі html сторінок розширення заборонено виконання inline скриптів (`<button onclick="...">`), вони повинні бути винесені у окремий файл, а всі події обробляються через `listener`, всі скрипти повинні бути завантажені локально. Наприклад:

```
<!doctype html>
<html>
  <head>
    <title>My Awesome Popup!</title>
    <script src="popup.js"></script>
  </head>
  <body>
    <button>Click for awesomeness!</button>
  </body>
</html>
```

Окремо слід розказати про використану систему передачі повідомлень між модулями розширення для їх комунікації. Зв'язок між розширеннями і їх сценаріями працює за допомогою передачі повідомлень. Кожна зі сторін може прослуховувати повідомлення, відправлені з іншого кінця, і реагувати на тому ж каналі. Повідомлення може містити будь допустимий об'єкт JSON (NULL, логічне, число, рядок, масив або об'єкт). Існує простий API для разового запиту і більш складне API, що дозволяє робити довготривалі з'єднання для обміну багатьма повідомленнями зі спільним контекстом. Крім того, можна відправити повідомлення на інше розширення якщо наявне його ID.

У роботі використовувались одноразові запити для з'єднання скриптів `background` та `popup`.

```
chrome.runtime.sendMessage({greeting: "hello"}, function(response) {
  console.log(response.farewell);
});
```

Серверна частина проекту працює на вбудованому у python CGI (common gateway interface) - стандарт інтерфейсу, який використовується для організації взаємодії програми веб-сервера із зовнішньою програмою. Сам інтерфейс розроблений таким чином, щоб можна було використовувати будь-яку мову програмування. Для обміну даними використовуються стандартні інтерфейси вводу/виводу.

Протокол обміну даними – HTTP POST (Передає призначені для користувача дані заданому ресурсу. Наприклад, в блогах відвідувачі зазвичай можуть вводити свої коментарі до записів в HTML-форму, після чого вони передаються серверу методом POST, і він поміщає їх на сторінку. При цьому передані дані включаються в тіло запиту. На відміну від методу GET, метод POST не вважається ідемпотентним, тобто багатократне повторення одних і тих же запитів POST може повертати різні результати.

Підсумовуючу слід сказати, що вибрана концепція клієнт-сервер є найбільш зручною, враховуючи той факт, що обробка даних займає процесорний час – потужний сервер з якісним розпаралелюванням дає змогу масштабувати проект. Кількість одночасних клієнтів теоретично сягає тисяч. На жаль, не можливо оцінити складність алгоритмів nlp у пакеті nltk та query (вони не описані), це дало б можливість кількісно оцінити час, затрачений на обробку одного результату.

3.2 Приклади роботи програми SQS

Сценарій взаємодії користувача та програми відбувається або через текстовий запит у роруп вікно (рис.17) або пошук виділеного тексту з веб сторінки через строку у контекстному меню (рис. 16). Перше дозволяє задавати звичайні природні питання, наприклад Who is Stephen Hawking? Where is Empire State Building? (всі можливі варіанти питання були розглянуті у попередніх розділах), не важко здогадатись, що це дозволяє шукати більш конкретні факти (списки, дати, географічні відомості). Другий же варіант запиту через виділення слова у тексті дозволяє шукати ключові слова у базах знань. Оскільки підтримується тільки англійська мова, то пакет NLTK з легкістю справляється з класифікацією всіх можливих форм англійських слів (відсутні відмінки, роди слова).



Рис.16 – скріншот роботи SQS

Розглянемо приклад пошукового запиту про невідомий нам термін Nauru (рис. 16-17) з попап та зі сторінки для 2 баз знань. Виберемо базу знань Freebase. Результатом буде наступний скріншот (рис. 18) – на затемненому фоні впливає форма з результатом пошуку, яка зникає після кліку курсором на фон.



Рис.17 – пошук через віконце роруп



Рис. 18 – Результат роботи SQS через Freebase

Далі спробуємо вибрати базу DBpedia. Результату запиту, на жаль, нема (рис.19). Дана ситуація можлива, адже не вся інформація є у структурованому вигляді у вибраних базах знань, саме тому реалізовано можливість вибору між базами Freebase, DBpedia.

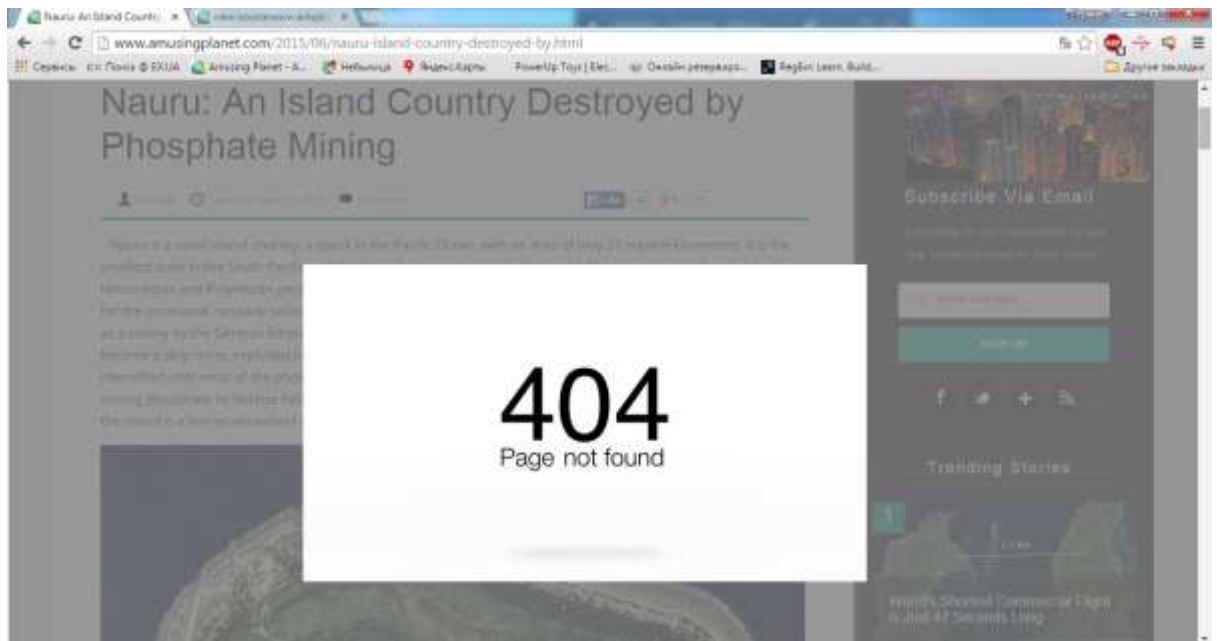


Рис. 19 – Результат не знайдено

Далі покажемо робочий запит до DBpedia. Пошуовим запитом буде інформація про музиканта John Lennon, результат запиту показано на (рис. 20). Відмінність інформації, що зберігається у цих двох базах добре видно на цьому прикладі – Dbpedia зберігає її у більш кластеризованому вигляді, тому запит не має картинки (для цього потрібно робити окремий запит).



Рис. 20 – Результат роботи SQS через DBpedia

Останнім прикладом показано можливість SQS виводити специфічні запити, як, наприклад, виведення списку акторів у фільмі. Наступний запит виводить головних акторів фільму «Поліцейська академія» – List actors of Police Academy (рис. 21)



Рис. 21 – List actors of Police Academy

В якості висновку слід відмітити, що SQS повністю виконує закладені в ньому функції та повністю відповідає вимогам, які були поставлені при проектуванні: так, основним досягненням у можна вважати, що користувач отримує шукану інформацію за 3 кліка курсором, уникаючи при цьому перегляд та аналіз звичайних пошукових результатів (наприклад Google).

3.3 Аналіз результатів роботи

Для початку згадаємо, що було поставлено за мету на початку розробки проекту SQS. Детально функції були описані в технічному завданні:

- Пошук виділеного слова або словосполучення по вибраній базі знань
- Трансформація ключового слова у SPARQL запит
- Можливість вибору бази знань: DBpedia та Freebase
- Вивід результату запиту у випадające вікно
- Пошук по ключовому слову за 3 кліка
- Можливість задати питання природньою мовою у рорир меню розширення

Ці вимоги були сформовані виходячи з того, що технологія зв'язаних даних позбавлена нормального графічного інтерфейсу, що унеможлиблює її використання непідготовленим користувачем.

SQS повністю відповідає заданим вимогам та реалізує концепцію унікального інтерфейсу для запитів до DBpedia, Freebase. Плагін розроблявся з можливістю подальшого масштабування – нові регулярні вирази для питань, нові області ключових слів та запити до ресурсів Open Linked Data в цілому. Для цього потрібно покращувати алгоритми відсіювання зайвих результатів та підвищувати швидкість генерації sparql запиту.

За допомогою сайту <http://www.websitepulse.com/help/tools.php> була проведена оцінка затраченого часу на отримання відповіді з серверу (рис. 22) та деталізація по кожному пункту (рис. 23) на питання What is the capital of Bolivia?

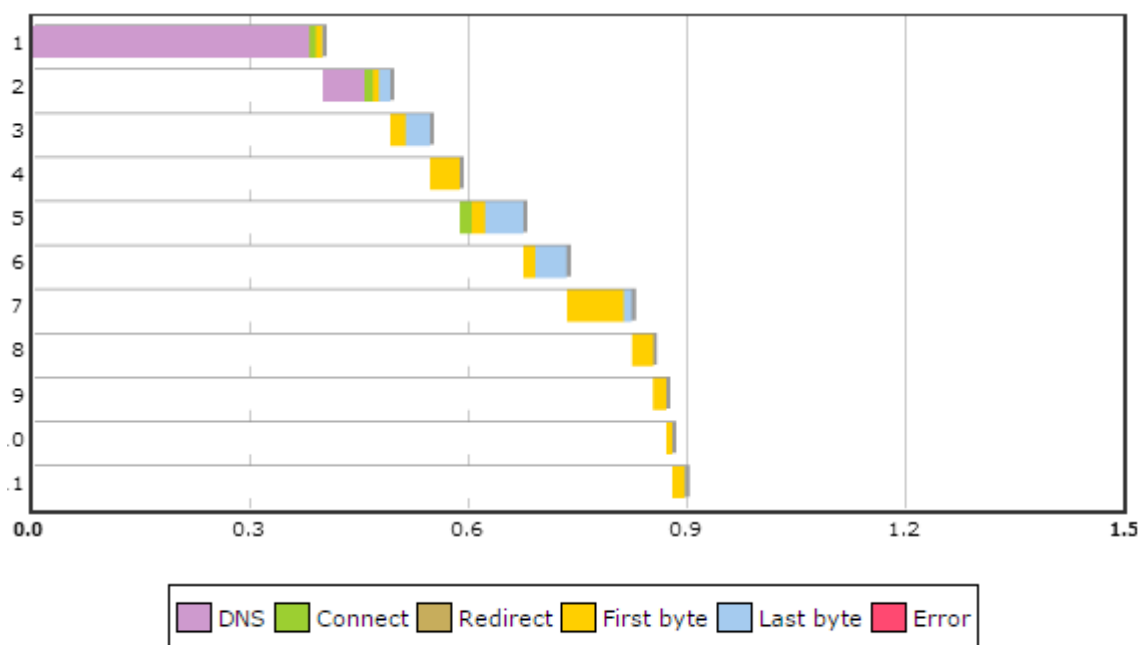


Рис. 22 – Діаграма завантаження модулів для SQS

#	Status	Time	DNS (sec)	Connect (sec)	Redirect (sec)	First (sec)	Last (sec)	Total (sec)	Size (Kb)
1	OK	12:17:36	0.3820	0.0086	0.0000	0.0086	0.0008	0.4000	10.65
2	OK	12:17:36	0.0575	0.0113	0.0000	0.0084	0.0157	0.0930	32.11
3	OK	12:17:36	0.0000	0.0000	0.0000	0.0210	0.0333	0.0543	95.01
4	OK	12:17:36	0.0000	0.0000	0.0000	0.0406	0.0001	0.0407	4.39
5	OK	12:17:36	0.0004	0.0162	0.0000	0.0183	0.0522	0.0871	91.44
6	OK	12:17:36	0.0000	0.0000	0.0000	0.0166	0.0431	0.0598	232.16
7	OK	12:17:36	0.0001	0.0000	0.0000	0.0780	0.0110	0.0890	27.08
8	OK	12:17:36	0.0000	0.0000	0.0000	0.0290	0.0000	0.0291	10.07
9	OK	12:17:36	0.0000	0.0000	0.0000	0.0182	0.0000	0.0182	1.37
10	OK	12:17:36	0.0000	0.0000	0.0000	0.0083	0.0000	0.0084	0.64
11	OK	12:17:36	0.0000	0.0000	0.0000	0.0170	0.0010	0.0181	13.31

Рис. 23- Затрачений час

Як бачимо на отримання відповіді потрібно менше 1 секунди і 3 кліка курсором, що навіть неможливо порівняти з загальноприйнятими пошуками у

Google та інших системах, адже вони не видають кінцевий результат. Єдиний конкурент SQS – Wolfram Alpha, система має набагато ширші можливості, але також має можливість пошуку по природнім запитах. Наступний графік показує затрачений час на отримання відповіді з цієї системи:

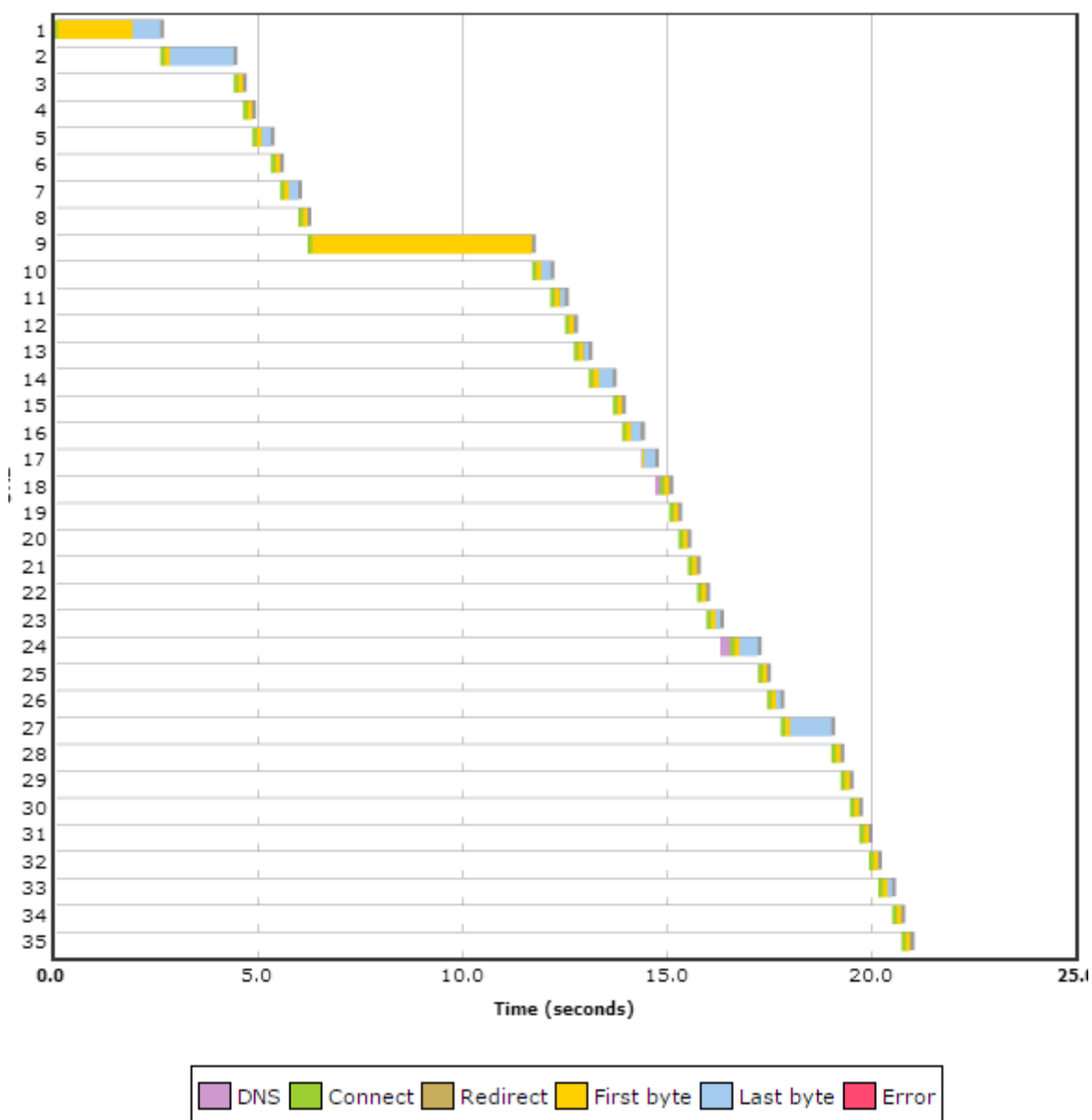


Рис. 24 – Діаграма завантаження модулів для Wolfram Alpha

Сумарно було витрачено більше 20 секунд часу на очікування результату, що більш ніж в 20 раз перевищує показник SQS! Але вольфрам показав набагато більше інформації по запиту: населення, карти, час, сусідні міста та ін. Можна сказати, що ми отримали взагалі всю відому інформацію по запиту про Болівію.

Це пояснює настільки довге очікування результату – він містить нерелевантну інформацію.

3.4 Висновок

Отже, робимо висновок – SQS має достойний результат та видає саме ту інформацію, що була запитана користувачем. Слід зробити знижку на те, що розширення не має достатньо користувачів та навантаження на сервер мізерне, але завдяки широким можливостям до масштабування ПЗ це буде легко вирішити.

4. ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1. Вступ та теоретичні відомості охорони праці

Далеко не останнім на сьогоднішній день є питання охорони праці та безпеки на робочому місці. Цей напрямок став настільки актуальним, тому що включає в себе всі аспекти робочого процесу: умови праці на робочому місці, безпеку технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови, що повинні відповідати вимогам нормативних актів про охорону праці. Все це робить людську працю захищеною на законодавчому рівні. В законі України «Про охорону праці» визначено, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Правильна організація трудової діяльності людини можлива лише при умові дотримання всіх заходів з охорони праці та наявності відповідних знань в області фізіології праці.

Всі, без виключень, підприємства та організації повинні створювати безпечні і нешкідливі умови праці. Забезпечення цих умов покладається на власника або уповноважений ним орган. Він не має права вимагати від працівника виконання роботи, поєднаної з явною небезпекою для життя, а також в умовах, що не відповідають законодавству про охорону праці. Працівник має право відмовитися від дорученої роботи, якщо створилася виробнича ситуація,

небезпечна для його життя чи здоров'я або людей, які його оточують, і навколишнього середовища.

Робочий процес повинен протікати у сприятливих умовах, це сприяє розвитку позитивних якостей працівника, дає широкі можливості для високопродуктивної і творчої роботи, сприяє зниженню аварійності та випадків виробничого травматизму. Тому охорона праці розглядається як одна з найважливіших економічних і соціальних задач не тільки окремого підприємства, але й держави в цілому.

4.2. Загальні вимоги до приміщення

Відповідальність по забезпеченню санітарно-побутових умов праці покладено на керівників структурних підрозділів.

Площу приміщень, в яких розташовують персональні комп'ютери, визначають згідно з чинними нормативними документами з розрахунку на одне робоче місце, обладнане ПК (табл. 4):

Таблиця 4 – Норми для робочого місця

Параметр	Норма	Примітка
Площа	Не менше 6 кв.м.	-
Об'єм	Не менше 20 куб.м.	З врахуванням макс. кількості осіб, які одночасно працюють
Відстань робочого місця від вікна	Не менше ніж 1 м.	Відстань рахують від вікна
Відстань між бічними поверхнями комп'ютерів	не менша за 1,2 м	-
Відстань між тильною поверхнею одного комп'ютера та екраном іншого	Не менша за 2,5 м	-
Прохід між рядами робочих місць	Не менший за 1 м	-

Інші вимоги до організації приміщення: заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками для запобігання випадкового дотику. В робочих приміщеннях повинні бути медичні аптечки першої допомоги та система автоматичної пожежної сигналізації з димовими пожежними сповіщувачами та переносними вуглекислотними вогнегасниками з розрахунку 2 шт. на кожні 20 кв.м площі приміщення. Підходи до засобів пожежогасіння повинні бути вільними.

4.3. Вимоги до організації робочого місця

Робоче місце працівника повинно забезпечувати оптимальну сидячу позу з такими характеристиками: ступні ніг - на підлозі або на підставці для ніг; стегна - в горизонтальній площині; передпліччя - вертикально; лікті - під кутом 70 - 90 град. до вертикальної площини; зап'ястя зігнуті під кутом не більше 20 град. відносно горизонтальної площини, нахил голови - 15 - 20 град. відносно вертикальної площини. Якщо користування ПК є основним видом діяльності, то ПК і його периферійні пристрої (принтер, сканер тощо) розміщується на основному робочому столі з лівого боку. Висота робочої поверхні столу для ПК має бути в межах 680 - 800 мм, а ширина - забезпечувати можливість виконання операцій в зоні досяжності моторного поля. Він повинен мати простір для ніг висотою не менше 600 мм, шириною не менше 500 мм, глибиною на рівні колін не менше 450 мм, на рівні витягнутої ноги – не менше 650 мм.

Робоче крісло користувача ПК повинно мати такі елементи: сидіння, спинку стаціонарні або знімні підлокітники.

Монітор та клавіатура мають розташовуватися на такій оптимальній відстані від очей користувача, але не повинні бути ближче ніж 600 мм, з урахуванням розміру алфавітно-цифрових знаків та символів.

Хибна організація робочого місця сприяє загальній і локальній напрузі м'язів шиї, тулуба, верхніх кінцівок, скривленню хребта й розвитку остеохондрозу та інших захворювань.

4.4. Аналіз параметрів робочого місця та приміщення

4.4.1. Геометричні параметри приміщення

В даному розділі проводиться аналіз середовища в якому розроблювався програмний продукт на основі санітарних норм України. Робоче приміщення (Табл. 5) розташоване на першому поверсі офісного будинку.

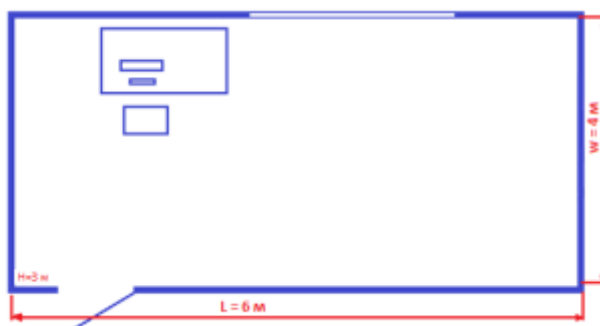


Рис.25 - План приміщення

Таблиця 5 – Параметри кімнати

Параметр приміщення	Значення
Довжина, м	6
Ширина, м	4
Висота, м	3
Площа, м ²	24
Об'єм, м ³	72

Відповідно до [29], площа S' , виділена для одного робочого місця з персональною ЕОМ, повинна складати не менше 6 кв. м, а об'єм V' – не менше 20 куб. м. В нашому випадку на одного працюючого виділене все приміщення, тобто 24 кв. м. і 72 куб.м.

Бачимо, що площа та об'єм приміщення відповідає нормам згідно [29].

4.4.2. Параметри робочого місця

Таблиця 6 - Порівняння фактичних і нормативних характеристик робочого місця

Параметр	Нормативне значення	Фактичне
Висота робочої поверхні	680-800 мм	700 мм
Глибина робочої поверхні	800-1000мм	900 мм
Висота сидіння над рівнем підлоги	400-500 мм	480 мм
Висота спинки стільця	300+/- 20 мм	320 мм
Регулювання нахилу спинки крісла	1-30°	1-30°
Глибина сидіння	400 мм та більше	500 мм
Висота простору для ніг	600 мм та більше	750 мм
Ширина простору для ніг	500 мм та більше	500 мм
Глибина простору для ніг	650 мм та більше	700 мм
Відстань від екрану до очей	600-700 мм	600 мм

Робочий стіл на досліджуваному місці також містить достатньо простору для ніг. Крісло, що використовується в якості робочого сидіння, є підйомно-поворотним, має підлокітники і можливість регулювання за висотою і кутом

нахилу спинки. Екран монітору знаходиться на відстані 0.8м, клавіатура має можливість регулювання кута нахилу 5-15° (табл. 6). Отже, за всіма параметрами робоче місце відповідає нормативним вимогам.

4.4.3. Аналіз та розрахунок стану освітлення в приміщенні

Відносно вікон робоче місце повинно бути розміщено так, щоб природне світло було збоку, переважно з лівого. Робоче місце, обладнане ПК повинно бути розташоване так, щоб уникнути попадання в очі прямого світла. Джерела штучного світла рекомендується розташувати з обох сторін від екрану паралельно напрямку зору. Вікна приміщень повинні мати регулювальні пристрої для відкривання. Як в приміщенні джерелом світла є штучне освітлення, то застосовуватися, як правило, люмінесцентні лампи.

Робота за дисплеєм ПЕОМ за розрядом зорових робіт відноситься до III розряду. При загальному висвітленні освітленість робочого місця повинна становити від 200 до 400 лк.

Розглянемо джерела природнього освітлення. В приміщенні знаходиться одне велике вікно з однієї сторони. Його характеристики (Табл. 7):

Таблиця 7 – Характеристика джерел освітлення

Параметр	Значення
Висота L	1.7 м.
Ширина W	2 м.
Відстань від підлоги	1 м.
Площа S	3.4 кв. м.

Вікна металопластикові, виходить на захід, можуть відкриватися та мають жалюзі.

Розглянемо джерела штучного освітлення. При штучному освітленні нормуються наступні параметри:

E (лк) - найменша припустима освітленість;

M - показник дискомфорту;

K_p (%) - коефіцієнт пульсації освітленості;

Перевіримо, чи відповідають нормам фактичні параметри штучного освітлення в приміщенні. Номінальний світловий потік лампи білого світіння ЛБ-40.

$$\Phi_n = 3120 \text{ лм.}$$

У приміщенні застосовуються світильники, у яких встановлені дві лампи.

Висоту підвісу світильника визначимо з формули :

$$h = H - h_c - h_p - h_n,$$

де

H - висота приміщення, м; h_c - висота світильника, м; h_n - відстань від стелі до підвісу, м; h_p - висота робочої поверхні, м.

Для розглянутого приміщення :

$$H = 2,7 \text{ м; } h_c = 0,2 \text{ м; } h_n = 0,3 \text{ м; } h_p = 0,7 \text{ м.}$$

звідси :

$$h = 2,7 - 0,2 - 0,3 - 0,7 = 1,5 \text{ м.}$$

Світильники розташовані в 4 ряди. Відстань між рядами 2 метра, відстань від ряду до стіни 0,5 метра. Приміщення має наступні габарити:

довжина $L = 6$ м.

ширина $W = 4$ м.

Визначимо освітленість у робочій точці. Для розрахунку загальної рівномірної освітленості при горизонтальній робочій поверхні використаємо метод коефіцієнта використання світлового потоку.

Розрахункова формула для світлового потоку світильника має вигляд:

$$\Phi_{\text{л}} = \frac{E \cdot K_3 \cdot S \cdot Z}{N \cdot n},$$

де

N - число світильників у приміщенні, $N = 4 \cdot 2 = 8$;

n - коефіцієнт використання світлового потоку;

$\Phi_{\text{л}}$ - світловий потік ламп;

K_3 - коефіцієнт запасу, $K_3 = 1.5$;

Z - коефіцієнт нерівномірності;

S - площа приміщення;

E - освітленість, створювана всіма світильниками.

Звідси одержуємо формулу для розрахунку освітленості на робочому місці

:

$$E = \frac{\Phi_{\text{л}} \cdot N \cdot n}{K_3 \cdot S \cdot Z};$$

Коефіцієнт використання світлового потоку залежить від:

ККД, кривій розподілу сили світла світильника;

Коефіцієнта відбиття стелі R_c і стін R_s ;

Висоти підвісу світильників $h_{\text{п}}$;

Показника приміщення і обчислимо за формулою:

$$i = \frac{L \cdot W}{h \cdot (L + W)};$$

$$i = (6 \cdot 4) / (1,5 \cdot (6 + 4)) = 1,6.$$

Нам відомо, що стеля й стіни пофарбовані в світло-сірий і світло-бежевий кольори. Приймаємо:

$$R_{\text{п}} = 50\%, R_{\text{с}} = 30\%.$$

$$\text{Звідси: } n = 42\%.$$

$$E = \frac{3120 \cdot 8 \cdot 0,42}{1,5 \cdot 24,94 \cdot 1,1} = 254,75 \text{ лк.}$$

Виходячи з всього вище згаданого фактична освітленість робочого місця повністю задовольняє вимогам і становить 254,75 лк. (при нормі від 200 до 400 лк.). Отже, для роботи з дисплеєм цілком достатньо існуючих джерел світла.

5.4.4. Аналіз повітряного середовища приміщення

Розглянемо основні параметри мікроклімату приміщення, які безпосередньо впливають на працівника.

Таблиця 8 - Значення мікроклімату

Період року	Параметр	Оптимальний	Фактичний
Теплий	Температура	23 – 25 °С	25-27 °С
	Вологість	40 – 60 %	50 %
	Швидкість повітря	≤ 0.1 м/с	
Холодний	Температура	22 – 24 °С	23-25 °С
	Вологість	40 – 60 %	50 %
	Швидкість повітря	≤ 0.1 м/с	

Всі показники задовольняють зазначеним вимогам для робіт категорії «легка Іа» і є задовільними для здоров'я людини. В приміщенні відсутні джерела шкідливих речовин. Спеціальні заходи з поліпшення або нормалізації цього параметру не потрібні. Умови мікроклімату у розглянутому приміщенні задовольняють вимогам встановленим у [1].

4.4.5. Виробничий шуму й вібрації

Для програмістів ПК допустимі значення звукового тиску та звуку на робочих місцях зазначені в таблиці 9

Таблиця 9 – Рівні допустимого шуму

Допустимі рівні звукового тиску (дБ) в стандартизованих октавних смугах з середньгеометричними частотами, Гц									Допустимий рівень звуку, дБА
32	63	125	250	500	1000	2000	4000	8000	
86	71	61	54	49	45	42	40	38	50

Джерелами шуми в робочому приміщенні є системи охолодження системного блоку ПК. По технічній документації сумарний шум не перевищує 32 дБА, що задовольняє нормативним вимогам. Вібрації відсутні.

Захист досягається в результаті відповідної організації роботи, складення і дотримання графіку, при якому час контакту з джерелом шуму не перевищують норму. Це дозволяє підтримувати здоров'я працівника та високу його продуктивність.

4.4.6. Вимоги до пожежної та електро безпеки

4.4.6.1 Вимоги електробезпеки

Вимоги до електромережі а також запобіжні засоби для уникнення травм від контакту з струмовідними елементами електроустаткування:

- величина напруги мережі не більше за 380В та 220В (міжфазна лінійна і фазна відповідно);
- всі струмовідні елементи (в першу чергу електричні дроти) вкриті ізоляційними матеріалами;
- в джерелі безперебійного живлення персонального комп'ютера використовується механічне захисне блокування, що забезпечує вимикання напруги при його відкриванні;
- електромережа в приміщенні розведена в спеціальних каналах стін і підлоги.

В заключення можна сказати, що приміщення задовольняє вимогам електробезпеки.

4.4.6.2 Пожежна безпека

З огляду на можливість виникнення пожежі слід з'ясувати, які речовини і матеріали можуть горіти. У приміщенні, що розглядається, можуть горіти вироби з дерева, пластмас, тканини і паперу. Тому приміщення, що аналізується, відноситься, відповідно до нормативної документації, до зони П-Па [8] і до категорії пожежної небезпеки В.

Ймовірними причинами виникнення пожежу можуть бути несправність електрообладнання (кабелів, розеток), короткі замикання внаслідок виходу з ладу чи експлуатації несправного електроустаткування (ПЕОМ, периферійних пристроїв), порушення правил протипожежної безпеки тощо.

Комплекс заходів для попередження пожеж:

- - обов'язковий інструктаж персоналу з питань охорони праці,
- - зокрема, правила пожежної безпеки у приміщеннях з ЕОМ;
- - заборона використання відкритого вогню у приміщенні;
- - розміщення схеми евакуації людей при пожежі і ознайомлення з нею персоналу.

Для гасіння пожежі кожна кімната обладнана ручними вуглекислотними вогнегасниками ВВК-1,4 [9]. Розглянуте приміщення обладнане датчиками централізованої системи пожежної сигналізації. Призначена відповідальна особа, що відповідає за дотримання персоналом вимог пожежної безпеки. Розроблено план евакуації персоналу і найбільш коштовного устаткування.

Отже, пожежна безпека відповідає нормам[9].

4.5. Висновок

В даному розділі „Охорона праці ” було проведено детальний структурний аналіз умов праці у приміщенні, де розробляється дипломний проект. Були наведені характеристики робочого приміщення та робочого місця, виявлена їх відповідність до норм. Проаналізовано стан освітленості приміщення (застосовується природне освітлення та штучне - два ряди світильників Л201Б 4x40-0.3, у кожному з яких знаходиться по дві лампи типу ЛБ-40). Характер робіт складності є допустимим рівнем напруженості і рекомендовано робити перерви по 10 хвилин після кожної години роботи. Встановлено, що в середньому температура повітря у приміщенні становить 24⁰ С. Зазначено, що приміщення за групою електробезпечності відноситься до приміщень без підвищеної небезпеки ураження струмом. Також визначені вогнегасники – ВВК-1,4 та ВВП.

В результаті були зроблені висновки про загальну відповідність умов праці нормативним вимогам.

ВИСНОВКИ

Мета дипломної роботи полягала в огляді технологій стеку семантичного веб, аналізі сучасного стану Linked Open Data з точки зору візуалізації запитів користувача і, як результат, представленні власного рішення питання доступу для звичайного користувача до ресурсів зі структурованою інформацією.

Для досягнення результатів були проведені наступні дослідження: розглянуто базові принципи та технології Linked Open Data, вибрані та порівняні найбільші бази знань, вивчено алгоритми обробки природної мови, у практичній частині було розроблено та описано проект, що реалізує природньо-мовний інтерфейс для доступу до інформації у базах знань LOD. В результаті виконаної роботи вдалось розробити концепцію ПЗ у вигляді розширення для браузеру Google Chrome, що повністю відповідає поставленим вимогам. Розглянемо детальніше результати дипломної роботи по окремим частинам:

1. Базові принципи LOD. Під час аналізу стеку технологій та стандартів семантичного веб, були отримані відомості про сучасний стан розвитку семантичного веб – кількість структурованої інформації тільки в LOD перевищила за 31 млрд триплетів, і наразі ця структура має доступ лише по запитам на мові SPARQL. Серед успішних проектів, які базуються на семантичному веб були виділені Google Knowledge Graph, Wolfram Alpha – вони повністю розкривають потенціал зв'язаних даних (відповіді на питання, виведення додаткової інформації до запиту, тобто вони самі генерують потрібну інформацію). Виходячи з цього було розроблено проект SQS, який слугує прошарком між базою знань та людиною. Наступним кроком було порівняння найбільших баз знань Wikidata, DBpedia, Freebase, в результаті було вирішено використовувати тільки останні 2 у розробці, оскільки перша є основою LOD, а друга має простий API для доступу до інформації.

2. Практична частина. Розроблено проект семантичного плагіна для браузеру, який перетворює звичайну природну мову на SPARQL запит. Для цього проект пройшов відповідні етапи проектування: планування, створення архітектури клієнт-сервер, вибір засобів розробки та тестування. Важливо відмітити, що основою плагіну є модуль аналізу природної мови *query*, який надав можливість швидко та ефективно перетворювати текст на SPARQL (парсинг, виділення частин мови за допомогою NLTK, зіставлення з регулярним виразом, генерація SPARQL). В результаті, готовий продукт дозволяє отримати відповідь не більше ніж на 3 кліка курсором та у порівнянні з Wolfram Alpha показав наступні результати: SQS видає єдиний результат, якщо відповідь наявна у базі, за 1с часу (час замірявся на локальному сервері), Wolfram Alpha за 22с дає відповідь, але при цьому містить ще велику кількість зв'язаних фактів, які не були запитані. Як бачимо концепція SQS має великий потенціал та достойних конкурентів, що в подальшому буде стимулювати розвиток додатку. Адже при розробці первинної архітектури були закладені можливості для подальшого масштабування проекту.

Підсумовуючи, слід сказати, що мета поставленої задачі досягнута і вирішена в повному обсязі. Виконана робота пропонує новий оптимальний підхід до візуалізації запитів до ресурсів LOD і це можна вважати важливим кроком вперед у розвитку концепції зв'язних даних та поширенні її серед звичайних користувачів.

Розроблений плагін – це корисний додаток, який може знайти популярність серед широкого загалу користувачів у повсякденному використанні, під час серфінгу у інтернеті. Подальший розвиток проекту вбачається у модернізації серверної частини для забезпечення безперебійної роботи для великої кількості клієнтів, доробка алгоритмів аналізу тексту та фільтрації результатів, перехід на запити до LOD в цілому, а не окремих його баз. Обов'язковим завданням на майбутнє також є інформаційна безпека ПЗ та перенесення на інші браузери.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Тім Бернерс Лі Семантичний Веб / Тім Бернерс Лі // Scientific America. – Division of Nature America, Inc – 2001 – №143 – С. 13-35.
2. John F. Sowa Semantic Networks / Stuart C. Shapiro // Encyclopedia of Artificial Intelligence. – University of A Coruña, Spain – 1997 – С. 56-58.
3. Dieter Fensel Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential / Dieter Fensel, Wolfgang Wahlster, Henry Lieberman, James Hendler // The MIT Press – 2002 - С. 25-30.
4. Florian Bauer Linked Open Data: The Essentials A Quick Start Guide for Decision Makers / Florian Bauer, Martin Kaltenböck // REEEP – 2006 – С. 12-15.
5. Jim Rapoza SPARQL Will Make the Web Shine. / Jim Rapoza // eWeek. – 2012 – С. 10-35.
6. Charniak, Eugene Statistical Techniques for Natural Language Parsing / Charniak // AI Magazine – 1997 - №18 - С.33–44.
7. Grune, Dick Parsing Techniques - A Practical Guide / Grune, Dick; Jacobs, Criel J.H. // Vrije Universiteit Amsterdam, Amsterdam, The Netherlands – 990 – С. 34 – 46.
8. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 (затверджено Постановою Головного державного санітарного лікаря України від 10.12.1998 р. № 7).
9. Санітарні норми мікроклімату виробничих приміщень : ДСН 3.3.6.042-99. – [Чинний від 2000-01-01]. – К. : МОЗ України, 2000. – 42 с. – (Національні стандарти України).
10. Природне і штучне освітлення : ДБН В.2.5-28:2015 – [Чинний від 2015-01-01]. – К. : Міністерство будівництва, архітектури та житлово-

- комунального господарства України, 2015. – 171 с. – (Національні стандарти України).
11. Санітарні норми виробничого шуму, ультразвуку та інфразвуку : ДСН 3.3.6.037-99. – [Чинний від 2000-01-01]. – К. : МОЗ України, 2000. – 37 с. – (Національні стандарти України).
 12. Правила охорони праці під час експлуатації електронно-обчислювальних машин : НПАОП 0.00.-1.31-10. – [Чинний від 2010-03-26]. – К. : Держнагляд охорони праці України, 2010. – 7 с. – (Національні стандарти України).
 13. Охорона праці в офісі. Вимоги до робочого місця офісного працівника – [Електронний ресурс] . - Режим доступу: <http://gc.ua/business-news/oxorona-praci-v-ofisi-vimogi-do-robochogo-miscya-ofisnogo-pracivnika/>
 14. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною безпекою. НАПБ Б.03.002-2007. (затверджено наказом МНС України від 03.12.2007 № 833)