

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

Навчально-науковий комплекс "Інститут прикладного системного аналізу"
(повна назва інституту/факультету)

Кафедра Системного проектування
(повна назва кафедри)

«На правах рукопису»
УДК _____

«До захисту допущено»

Завідувач кафедри

_____ (підпис)

_____ (ініціали, прізвище)

“ _____ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності _____ 8.05010102 Інформаційні технології проектування

(код і назва спеціальності)

на тему: _____ Семантичні композитні веб-додатки в наукових дослідженнях

Виконав: студент 6 курсу, групи ДА-31м
(шифр групи)

_____ Вініченко Андрій Миколайович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник _____ ст.вик. Булах Б.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

_____ (підпис)

Консультант Охорона праці _____ к.б.н., доц. Гусєв А.М.

(назва розділу)

(науковий ступінь, вчене звання, прізвище, ініціали)

_____ (підпис)

Рецензент _____

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2015 року

**Національний технічний університет України
«Київський політехнічний інститут»**

Факультет (інститут) ННК «Інститут прикладного системного аналізу»
(повна назва)

Кафедра Системного проектування
(повна назва)

Освітньо-кваліфікаційний рівень «Магістр»
(назва ОКР)

Напрямок підготовки 6.050101 Комп'ютерні науки
(код і назва)

Спеціальність 8.05010102 Інформаційні технології проектування
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

А.І.Петренко
(ініціали, прізвище)

(підпис)

«__» _____ 2015 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Вініченку Андрію Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації: Семантичні композитні веб-додатки в наукових дослідженнях

науковий керівник дисертації ст.вик. Булах Богдан Вікторович,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «13» лютого 2015 р. № 19/1-ст

2. Строк подання студентом дисертації 08.06.2015

3. Об'єкт дослідження: Семантичні веб-додатки, використання онтологій, мешапи

4. Предмет дослідження: Застосування сем. додатків для наукових досліджень, Методи і принципи використання онтологій

5. Перелік завдань, які потрібно розробити:

1. Провести аналіз поняття композитних веб-сервісів(мешапів).
2. Дослідити принципи, мету, технічну сторону семантичних веб-додатків, використання і значення онтологій.
3. Запропонувати шляхи використання семантичних веб-додатків у наукових дослідженнях і дослідити існуючі.

4. Експериментально показати роботу семантичного веб-додатку.

6. Орієнтовний перелік ілюстративного матеріалу: презентація по темі «Семантичні композитні веб-додатки в наукових дослідженнях»

7. Орієнтовний перелік публікацій

Вініченко, А.М. Семантичні композитні веб-додатки в наукових дослідженнях [Текст] / А.М. Вініченко // Системний аналіз та інформаційні технології: матеріали 17-ї Міжнародної науково-технічної конференції SAIT 2015, Київ, 22-25 червня 2015 р. – 2015. – с.148

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	доц., к.б.н. Гусев А. М		
Основна частина	ст.вик. Булах Б.В.		

9. Дата видачі завдання 27.09.2014

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Отримання завдання	27.09.2014	
2	Збір інформації	19.01.2015	
3	Аналіз вимог завдання, вибір методів і засобів розв'язання поставленої задачі	15.02.2015	
4	Проведення аналізу структури і використання композитних веб-сервісів(мешапів)	13.03.2015	
5	Дослідження принципів, мети, технічною сторони семантичних веб-додатків, використання і значення онтологій	20.04.2015	
6	Пошук шляхів використання семантичних веб-додатків у наукових дослідженнях і аналіз існуючих	9.05.2015	
7	Навантажувальне тестування створеної системи та надання практичних рекомендацій	31.05.2015	
8	Розробка розділу з охорони праці	10.05.2015	
9	Оформлення дипломної роботи	9.06.2015	
10	Отримання допуску до захисту та подача роботи в ДЕК	12.06.2015	

Студент _____
(підпис)

А.М. Вініченко
(ініціали, прізвище)

Керівник роботи _____
(підпис)

Б.В. Булах
(ініціали, прізвище)

РЕФЕРАТ

на магістерську дисертацію

виконану на тему: Семантичні композитні веб-додатки в наукових дослідженнях

студентом: Вініченком Андрієм Миколайовичем

Робота виконана на 90 сторінках, містить 12 ілюстрацій. При підготовці використовувалася література з 30 різних джерел.

Актуальність

На сьогодні, збір фактів, побудова моделі предметної області, співвіднесення власної моделі з моделями інших дослідників являються типовими компонентами роботи дослідника, що займається аналізом предметної області якоїсь галузі науки. Всі завдання розвитку інформаційних технологій в науці з'являються від проблем збору та організації доступу до масивів даних, труднощів розвитку та роботи з комплексними моделями даних, необхідністю обчислень (що здійснюються за допомогою комп'ютерних систем високої продуктивності або за рахунок розподіленої мережі обчислень) і розуміння того, що рішення багатьох наукових проблем знаходиться на стику наукових дисциплін.

Наприклад, одним з таких завдань є знаходження нових фактів та інформації (оптимальних рішень) на основі існуючих даних. Вирішити дану проблему може використання семантичних веб-додатків.

В основі таких додатків лежать бази знань, які в свою чергу реалізуються за допомогою онтології. Підтримка семантичних зв'язків явним чином призводить до утворення багатосарової семантичної структури контенту електронної бібліотеки, кожен шар якої відповідає деякій властивості семантичних зв'язків. Така структура може формуватися на основі онтології зв'язків і може служити джерелом інформації для проведення якісно нових

наукометричних вимірювань, для дослідження структурних властивостей корпусу знань у різних галузях науки.

В останні роки на основі бібліографічних посилань у публікаціях, що випускаються в авторитетних періодичних виданнях, почали створюватися індекси цитування, які формують бібліометричну статистику. Зв'язки цитування в текстових публікаціях зазвичай представляються неструктурованим чином у вигляді списку використаної літератури. Поряд зі зв'язками цитування існують різноманітні інші семантичні зв'язки, наприклад, по результатам дослідження. Усі ці можливі семантичні зв'язки можуть бути використані в семантичних веб-сервісах, за допомогою яких можна буде знаходити нові дані чи значно продуктивніше шукати існуючі.

Мета і завдання

Метою магістерської дисертації є дослідження і аналіз композитних семантичних веб-додатків, визначення способів використання їх у наукових дослідженнях, реалізація прототипу такої системи для кращого розуміння і порівняння з технологіями, які використовуються зараз. Для цього визначено завдання, які вирішуються в роботі:

1. Провести аналіз поняття композитних веб-сервісів(мешапів).
2. Дослідити принципи, мету, технічну сторону семантичних веб-додатків, використання і значення онтологій.
3. Запропонувати шляхи використання семантичних веб-додатків у наукових дослідженнях і дослідити існуючі.
4. Експериментально показати роботу семантичного веб-додатку.

Рішення поставлених завдань та досягнуті результати

В роботі було розглянуто технології композитних семантичних веб-додатки, можливості їх використання в наукових дослідженнях. Були роз'яснені такі поняття як композитні веб-сервіси(мешапи), семантичний веб-онтології, бази знань, технології реалізації семантичних додатків. Під час

пошуку інформації та розуміння концепції семантичних веб-додатків було створено власну програму з метою ознайомлення з принципами, методиками і технологіями розробки таких веб-додатків. Було реалізовано приклади використання декількох точок доступу SPARQL для отримання інформації з онтологій та RDF-файлів в залежності від запитів SPARQL. Були використані бібліотеки для роботи з SPARQL запитами і RDF-файлами. Практична реалізація семантичного додатку забезпечило краще розуміння необхідності використання такої концепції побудови інформації. Такий підхід організації даних та доступу до них дає більше можливостей ніж поширена зараз конструкція СУБД + інтерфейс.

Об'єкт досліджень

У відповідності до поставленої мети об'єктом досліджень обрано концепцію використання семантичних веб-додатків, онтологій, принципи побудови мешапів, застосування у наукових дослідженнях.

Предмет досліджень

Предметом досліджень обрано методи і принципи використання онтологій, RDF-файлів, мову запитів до таких файлів SPARQL у веб-додатках, вебу в цілому.

Методи досліджень

Пошук і аналіз інформації про роботу семантичних веб-додатків, про Семантичний Веб, про семантичні зв'язки в наукових статтях. Розробка прототипу веб-додатку з використанням запитів до бази знань, пошуку в онтологіях з використанням існуючих бібліотек.

Наукова новизна

Наукова новизна роботи полягає в пропозиції способів використання семантичного вебу в наукових дослідженнях, виявлення нових семантичних

зв'язків в наукових статтях.

Практична цінність

В роботі було створено власну програму з метою ознайомлення з принципами, методиками і технологіями розробки композитних семантичних веб-додатків. Було реалізовано приклади використання декількох точок доступу SPARQL для отримання інформації з онтологій та RDF-файлів в залежності від запитів SPARQL. Також запропоновані способи використання таких додатків в наукових дослідженнях

Ключові слова

Семантичний веб, семантичні веб-додатки, композитні веб-додатки, мешап, онтології, SPARQL, RDF.

ABSTRACT

on master's thesis

on topic: Semantic Web applications mashup in research

Student: Andrey N. Vinichenko

Work carried out on 90 pages containing 12 figures. The paper was written with references to 16 different sources.

Topicality

Today, the collection of facts, subject area model construction, correlation own models with models of other researchers are typical components of the researcher, which analyzes the subject area of a science. All tasks of development of information technologies in science emerging from problems of collecting and providing access to data sets, the difficulties of working with complex data models, the need for calculations (carried out through a computer system performance or through a distributed network computing) and understanding that the solution of many scientific problems is at the intersection of disciplines.

For example, one of these tasks is to find new facts and information (optimal decisions) based on existing data. Solve this problem can use semantic web applications.

The basis of such applications are the knowledge base, which in turn implemented using ontology. Support of semantic relations explicitly leads to the formation of a multilayer structure semantic content of electronic library, each layer of which corresponds to some properties of semantic links. This structure can be formed based on ontology and may be a source of information for qualitatively new scientometric measurements to study the structural properties of the hull knowledge in various fields of science.

In recent years, based on bibliographic references of publications produced in reputable journals, citation indexes were established that form bibliometric statistics. Links citation in text publications are usually presented unstructured way in a list of references. Along with links are citing various other semantic relationships, such as on the results of the study. All these possible semantic connections can be used in semantic web service through which you can find new information or seek much more productive ones.

Aims and objectives

The purpose of the master's thesis is the study and analysis of composite semantic web applications, identify ways of using them in research, prototype implementation of the system for better understanding and comparison with technologies that are used now. To do this, set tasks are solved in the work:

1. To analyze the concept of composite web services (mashups).
2. To study the principles, objectives, technical side Semantic Web applications, use and meaning of ontologies.
3. Suggest ways of using semantic web applications in research and investigate existing.
4. Experimentally show the work of the semantic web applications.

The solution of the tasks and results

The work was considered technology of Semantic Web mashup applications, their possible use in research. There were explaining concepts such as composite Web services (mashups), semantic web, ontologies, knowledge bases, technologies of semantic applications. It was created its own program to learn about the principles, techniques and technologies development of such web applications. It was realized examples of using some SPARQL end points for getting information from ontologies

and RDF-files, depending on SPARQL requests. It was used the library to work with SPARQL queries and RDF-files. Feasibility semantic application provided a better understanding of the need for such a concept of building information. This approach data organization and access gives more than is common database design + interface.

The object of research

Pursuant to the target object is selected concept studies using semantic web applications, ontologies, mashups construction principles , use in scientific research.

Subject of research

The subject of research was chosen methods and principles using ontologies, RDF-files, query language SPARQL to such files in web applications, web in general.

Research Methods

Search and analysis of information about the semantic web applications work, the Semantic Web, the Semantic relations in scientific articles. Development of a prototype web application using queries to the knowledge base, search in ontologies using existing libraries.

Scientific novelty

The scientific novelty of this work is to offer ways to use the Semantic Web to research, identify new semantic relations in scientific articles.

The practical value of research

In this work was created its own program to get acquainted with the principles, methods and technology for developing semantic web mashup applications. It was realized examples of using some SPARQL end points for getting

information from ontologies and RDF-files, depending on SPARQL requests. Also proposed methods of using such applications in research.

Keywords

Semantic Web, Semantic Web applications, composite web applications, mashup, ontologies, SPARQL, RDF.

ЗМІСТ

ЗМІСТ	12
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ.	14
ВСТУП	15
1 КОМПОЗИТНІ СЕМАНТИЧНІ ВЕБ-ДОДАТКИ.....	18
1.1 Композитні веб-додатки.....	18
1.2 Типи мешапів	19
1.3 Переваги використання композитних додатків.....	21
1.4 Архітектурні аспекти мешапів	21
1.5 Семантичні веб-додатки.....	22
1.6 Перспективи використання семантичних веб-додатків.....	26
1.7 Композитні семантичні веб-додатки, задачі композиції веб-сервісів у семантичному веб просторі.....	28
1.8 Висновки.....	32
2 БАЗИ ЗНАНЬ. ОНТОЛОГІЇ. СЕМАНТИЧНІ ЗВ'ЯЗКИ У НАУКОВИХ ДОСЛІДЖЕННЯХ	33
2.1 Семантичний Web.....	33
2.2 Список основних діючих рекомендацій W3C, пов'язаних з Семантичним Web	34
2.3 Структура семантичного Web	35
2.4 Засоби опису ресурсів RDF	38
2.5 Онтології.....	40
2.6 Знання	43
2.7 Агенти	44
2.8 Виявлення сервісів.....	45
2.9 Реалізації семантичного WEB	48
2.10 Використання семантичного WEB в наукових дослідженнях.....	49
2.11 Висновки.....	50
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СЕМАНТИЧНОГО ВЕБ-ДОДАТКУ	52
3.1 Вибір технологій	52

	13
3.2 Архітектура додатку.	53
3.3 Інформація про використані бібліотеки	55
3.4 Функціонал створеного додатку.....	61
3.5 Висновки	67
4 ОХОРОНА ПРАЦІ	68
4.1 Характеристика робочого місця	68
4.2 Аналіз умов праці.....	69
4.2.1 Напруженість праці програміста	69
4.2.2 Мікроклімат	70
4.2.3 Шум	71
4.2.4 Освітлення робочого місця	72
4.2.5 Електробезпека	73
4.2.6 Пожежна безпека.....	73
4.2.7 Випромінювання	74
4.2.8 Ергономіка робочого місця	74
4.2.9 Інструкція з техніки безпеки.....	75
4.3 ВИСНОВКИ.....	76
ВИСНОВКИ.....	77
ПЕРЕЛІК ПОСИЛАНЬ	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ.

CRUD — (англ. create read update delete) 4 базові функції управління даними «створення, зчитування, зміна і видалення».

REST (скор. англ. Representational State Transfer, «передача стану подання») — підхід до архітектури мережових протоколів, які забезпечують доступ до інформаційних ресурсів.

SPARQL - стандартна мова запитів до даних у форматі RDF (Protocol and RDF Query Language).

OWL – мова, створена для вираження онтологій (Ontology Working Language).

Т. д. SPARQL - точка доступу SPARQL, служба, що підтримує протокол запитів SPARQL.

Мешапи (Mashup) - веб-сторінка або веб-додаток, який використовує контент з більш ніж одного джерела, щоб створити єдиний новий сервіс, що використовує єдиний графічний інтерфейс.

XMLHttpRequest — API-запит веб-клієнта (браузера) до веб-сервера за протоколом HTTP у фоновому режимі, для мов програмування JavaScript, JScript, VBScript і подібних. Дозволяє здійснювати HTTP-запити до віддаленого сервера без потреби перезавантажувати сторінку.

Інтероперабельність - це здатність продукту або системи, інтерфейси яких повністю відкриті, взаємодіяти та функціонувати з іншими продуктами або системами без будь-яких обмежень доступу і реалізації.

ВСТУП

Типовими компонентами роботи дослідника, що займається аналізом предметної області, є збір фактів, побудова моделі предметної області, співвіднесення власної моделі з моделями інших дослідників. Вимоги до збору та організації доступу до масивів даних, труднощі розвитку та роботи з комплексними моделями даних, необхідність обчислень (що здійснюються за допомогою комп'ютерних систем високої продуктивності або за рахунок розподіленої мережі обчислень) і розуміння того, що рішення багатьох наукових проблем знаходиться на стику наукових дисциплін, ставить завдання розвитку інформаційних технологій в науці.

Одне з таких завдань – це знаходження нових фактів та інформації (оптимальних рішень) на основі існуючих даних. Використання семантичних веб-додатків може допомогти вирішити дану проблему.

Тому метою даної магістерської дисертації є дослідження переваг та аналіз способів побудови композитних семантичних веб-додатків, визначення можливих способів використання їх у наукових дослідженнях, реалізація прототипу такої системи для кращого розуміння переваг та недоліків технологій, які використовуються зараз.

Для цього визначено такі завдання, які вирішуються в роботі:

- Провести аналіз концепції композитних веб-сервісів(мешапів).
- Дослідити принципи, мету, технічну сторону семантичних веб-додатків, використання і значення онтологій.
- Запропонувати шляхи використання композитних семантичних веб-додатків у наукових дослідженнях, дослідити існуючі рішення.
- Експериментально перевірити можливість реалізації досліджених рішень на прикладі тестового семантичного веб-додатку.

Протягом дослідження проаналізовані кілька сценаріїв застосування семантичних додатків в наукових дослідженнях. В основі таких додатків лежать бази знань, які в свою чергу реалізуються за допомогою онтології. Бази знань наукових досліджень будуються на основі різних зв'язків. Підтримка семантичних зв'язків явним чином призводить до утворення багатошарової семантичної структури контенту електронної бібліотеки, кожен шар якої відповідає деякій властивості семантичних зв'язків. Така структура може формуватися на основі онтології зв'язків і може служити джерелом інформації для проведення якісно нових наукометричних вимірювань, для дослідження структурних властивостей корпусу знань у різних галузях науки.

В останні роки на основі бібліографічних посилань у публікаціях, що випускаються в авторитетних періодичних виданнях, почали створюватися індекси цитування, які формують бібліометричну статистику. Зв'язки цитування в текстових публікаціях зазвичай представляються неструктурованим чином у вигляді списку використаної літератури.

Поряд зі зв'язками цитування між документами наукових електронних бібліотек існують різноманітні інші семантичні зв'язки. Наприклад, зв'язок може вказувати, що її цільовий документ містить наукові результати, що базуються на результатах, описаних у вихідному документі зв'язку, або що у вихідному документі зв'язку спростовується результат, викладений у її цільовому документі. Зв'язок може також вказувати, що вихідний її документ є новою редакцією цільового документа або що він являє собою його складову частину, наприклад, анотацію. Існує велика різноманітність семантичних зв'язків, які можна при необхідності підтримувати між документами бібліотеки.

В результаті виконаних досліджень проаналізовано процеси пов'язані з науковою діяльністю, що спричиняють виникнення потреби формування семантичних зв'язків між публікаціями, а в середовищі електронної бібліотеки – між представленими в ній інформаційними об'єктами. Аналізуються також категорії семантичних зв'язків, які при цьому породжуються, розглянуто

властивості структури семантичних зв'язків, її багат шаровий характер, семантика зв'язків, питання їх класифікації, відомі онтології, створені для цих цілей, і онтологія, що використовується в прикладі.

Використання онтології ґрунтується на таких її характеристиках як:

- Децентралізація – можливість роботи з базою знань декількох сайтів на одній онтології
- Логічний висновок – можливість породження нових фактів
- Природне спадкування і вкладеність
- Опис предметної області, метадані

Композиція Веб-сервісів викликає значний інтерес у контексті проблематики семантичної Веб-мережі. Її цілі полягають у тому, щоб зробити існуючий Веб більш ефективним і забезпечити гнучкий підхід до управління всіма типами процесів. Очікувані програмні додатки, побудовані на основі композиції сервісів, зроблять більш перспективними такі напрямки як Веб-комерція, електронний уряд, електронна наука та інші.

1 КОМПОЗИТНІ СЕМАНТИЧНІ ВЕБ-ДОДАТКИ

1.1 Композитні веб-додатки.

Під композитними додатками у веб-розробці розуміють веб-сторінку або веб-додаток, який використовує контент з більш ніж одного джерела, щоб створити єдиний новий сервіс, що використовує єдиний графічний інтерфейс. Наприклад, ви могли б об'єднати ваш блог з сервісом аналітики відвідувань чи сервісом відправки електронних листів. Термін композитні веб-додатки означає легку, швидку інтеграцію, часто використовуючи прикладні програмні інтерфейси (відкрите API) та джерела даних для отримання збагачених результатів. Термін mashup(композитні) родом з Англії. В основному використовується по відношенню до музики, де музиканти об'єднують аудіо треки від різних пісень, таким чином, змушуючи їх разом, щоб створити щось нове.

Основні характеристики такого 'змішування' веб-додатків є комбінація, візуалізація та агрегація. Важливо, зробити з існуючих даних більш корисні дані, для особистого і професійного використання. Для того, щоб мати постійний доступ до даних інших послуг, такі додатки, як правило, є клієнтськими програми або веб-додатками.

В останні роки все більше і більше веб-додатків опублікували API, які дозволяють розробникам програмного забезпечення легко інтегрувати дані і функції, як SOA, а не будувати їх самостійно. Вважається, що композитні програми відіграють активну роль в еволюції соціального програмного забезпечення і Web 2.0. Інструменти використання таких програм, як правило, досить прості для використання кінцевими користувачами. Вони, як правило, не вимагають навичок програмування і мають досить зрозумілий візуальний зв'язок GUI віджетів, сервісів і компонентів разом. Таким чином, ці

інструменти сприяють новому баченню веб-мережі, де користувачі можуть внести свій вклад.

Історія мешапів починається від першого розуміння більш широкого контексту історії в Інтернеті. Для Web 1.0 бізнес-моделі компанії зберігали дані споживачів на порталах і оновлювали їх регулярно. Вони контролювали всі дані споживача, а споживач повинен був використовувати їх продукти і послуги, щоб отримати інформацію.

З появою Web 2.0 використання нових Web-стандартів дозволило розблокувати дані споживачів. У той же час, використання мешапів дозволило змішувати API конкурентів, тим самим створювати нові послуги.

Перші композитні програми використовували послуги відображення інформації або фото послуги, щоб об'єднати ці послуги з даними будь-якого роду і, отже, створити візуалізацію даних. На початку, більшість мешапів були орієнтовані на споживачів, але останнім часом такі додатки слід розглядати як цікава концепція що корисна також і для підприємств. Бізнес мешапи можуть об'єднати існуючі внутрішні дані з даними і можливостями зовнішніх послуг, щоб створити нові погляди на свої дані.

1.2 Типи мешапів

Існує багато різних типів мешапів таких як бізнес-мешапи, споживчі, і мешапи даних. Найбільш поширеним типом є споживчі мешапи, спрямовані на широку публіку.

- 1) Бізнес мешапи це такі додатки, які поєднують власні ресурси і дані з ресурсами і даними зовнішніх веб-сервісів. Вони зосереджують дані в одному представленні і дозволяють спільні дії між підприємствами і розробниками. Це добре працює для гнучкої розробки проекту, який вимагає співпраці між розробниками і замовником для визначення та реалізації бізнес-вимог. Бізнес мешапи добре захищені, мають

приємний зовнішній вигляд, додатки які надають корисну інформацію з різних внутрішніх і зовнішніх джерел інформації.

- 2) Споживчі мешапи об'єднують дані з декількох публічних джерел в браузері і організуються за допомогою простого інтерфейсу браузера (наприклад: Wikipediavision поєднує Google Map API і Вікіпедію).
- 3) Мешапи даних, на відміну від споживчих мешапів, об'єднують подібні типи медіа-контенту і інформації з декількох джерел в єдине представлення. Поєднання всіх цих ресурсів створює новий і відмінний веб-сервіс, який спочатку не був представлений належним чином.

За типом API композитні додатки можуть бути класифіковані в залежності від виконуваних кожним окремим сервісом функцій і даних, які надаються.

Типи даних:

- Індексовані дані (документи, блоги, зображення, відео, статті, торговельні робочих місць ...)
- Картографічні та географічні дані: ПЗ геолокації, geovisualization
- Feeds, подкасти: агрегатори новин

Функції:

- Конвертори даних: перекладачі, обробка голосу, URL shorteners ...
- Зв'язок: електронна пошта, обмін миттєвими повідомленнями, ...
- Обробка візуальних даних: візуалізація інформації, схеми
- Безпека зв'язків: електронні платіжні системи, ідентифікація ID ...
- Редактори

1.3 Переваги використання композитних додатків

Композитні додатки мають ряд переваг в порівнянні з використанням, наприклад, порталів. Мешапи використовують нові, слабо визначені "Web 2.0" методи. Використовується API, пропонуване сайтами з різним типом інформації для збору та повторного використання контенту в новому сенсі. Може працювати на чистому XML, а також на презентаційно-орієнтованому контенті (наприклад, HTML). Агрегація контенту може відбуватися або на сервері, або на клієнті. Об'єднання і презентація даних залежить тільки від сервісу який виводить їх користувачу. CRUD операції будуються на основі REST архітектурних принципів, але ніякого формального API не існує.

1.4 Архітектурні аспекти мешапів

Архітектура мешапів розділена на три шари:

- Презентація / взаємодія користувача: це інтерфейс користувача. Технології, що використовуються: HTML / XHTML, CSS, JavaScript, Asynchronous JavaScript і XML (Ajax).
- Web Services: функціональність продукту можна отримати за допомогою послуги API. Технології, що використовуються: XMLHttpRequest, XML-RPC, JSON-RPC, SOAP, REST.
- Дані: обробка даних, як відправка, зберігання та отримання. Технології, що використовуються: XML, JSON, KML.

Архітектурно є два стилі мешапів: на основі вебу і на основі сервера. У той час як мешапи на веб-основі, як правило, використовують веб-браузер користувача, щоб об'єднати і переформатувати дані, серверні мешапи аналізують і форматують дані на віддаленому сервері і передають дані в браузер користувача в його остаточному вигляді.

Композитні програми являють собою варіацію шаблону проектування Фасад. Шаблон розробки програмного забезпечення, який забезпечує спрощений інтерфейс для більшої частини коду (у даному випадку код агрегується з різних каналів з різними API).

Мешапи можуть бути використані у вигляді 'Програмне забезпечення як послуга' (SaaS).

Після декількох років розробки стандартів, основні підприємства починають впроваджувати сервіс-орієнтовані архітектури (SOA) для інтеграції розрізнених даних за рахунок їх надання в якості дискретних веб-служб. Веб-служби забезпечують відкриті, стандартизовані протоколи для забезпечення уніфікованих засобів доступу до інформації з широкого набору платформ (операційних систем, мов програмування, додатки). Ці веб-служби можуть бути повторно використані, щоб забезпечити абсолютно нові послуги та програми в межах і між організаціями, забезпечуючи гнучкість бізнесу.

1.5 Семантичні веб-додатки

Сьогодні через величезні розміри, невпорядкованість та слабоструктурованість Веб-контенту результати пошуку необхідної інформації у Вебі засобами глобальних пошукових системи не завжди можна вважати задовільними і такий пошук часто вимагає значних витрат часу користувача. Разом з цим, створення ефективних автоматичних пошукових засобів ускладнюється тим, що Веб був створений для опрацювання людьми, а не програмними агентами (структура Вебу, зокрема HTML-код Веб-сторінок, має дуже обмежені засоби класифікації частин контенту, хоча дозволяє здійснювати його багате візуальне представлення).

“Семантичний Веб” (“Semantic Web”) – термін, введений винахідником Вебу Тімом Бернерсом-Лі (Tim Berners-Lee). Семантика – це вивчення значень. Семантичний Веб – це підхід до розвитку Вебу, який полягає у збагаченні Веб-

контенту спеціалізованими мета-даними і створенні засобів для автоматичної обробки цих мета-даних (інтелектуальних програмних агентів), які могли б розпізнавати значення документів з метою виконання складних пошукових завдань користувачів.[1]

Основна ідея цього полягає в організації такого подання даних у мережі, щоб допускалася не тільки їх візуалізація, але і їх ефективна автоматична обробка програмами різних виробників. Шляхом таких радикальних перетворень концепції вже традиційного Web передбачається перетворення його в систему семантичного рівня. За задумом творців Семантичний Web повинен забезпечити "розуміння" інформації комп'ютерами, виділення ними найбільш придатних за тими чи іншими критеріями даних, і вже після цього - надання інформації користувачам.

При автоматичній обробці інформації в рамках Семантичного Web сервіси, що взаємодіють один з одним на основі аналізу смислових зв'язків між об'єктами і поняттями, що зберігаються в Мережі повинні відбирати лише ту інформацію, яка буде реально корисна користувачам.[1]

За визначенням консорціуму W3C Семантичний Web являє собою розширення існуючої мережі Internet, в якому інформація представляється в чіткому і певному смислового значенні, що дає можливість людям і комп'ютерам працювати з більш високим ступенем взаєморозуміння і узгодженості. Семантичний Web передбачає об'єднання різноманітних видів інформації в єдину структуру, де кожному смислового елементу даних буде відповідати спеціальний синтаксичний блок (тег). Теги повинні складати єдину ієрархічну структуру, на основі якої і повинен функціонувати Семантичний Web. За словами Бернерса-Лі, в рамках проекту "Семантичний Web" розробляються мови для вираження інформації у формі, доступній для машинної обробки, на яких можна буде описувати як дані, так і принципи трактування цих даних. Це повинно привести до того, що правила висновків,

що існують у якій-небудь одній системі подання знань, будуть передаватися по мережі іншим подібним системам. У процесі реалізації концепції Семантичного Web отримали широкий розвиток синтаксичні методи представлення інформації мовними засобами XML і його додатків, призначених для опису типових властивостей елементів XML- документів, їх структури і семантики: рекомендації W3C, що регламентують DTD (Document Type Definition), XML Schema, XQuery (мова запитів до баз XML-даних) і т.д.

Інша гілка Семантичного Web пов'язана з напрямками, близькими до галузі штучного інтелекту, і названа онтологічним підходом. Цей підхід включає в себе засоби анотування документів, якими могли б скористатися комп'ютерні програми - Web-сервіси та агенти при обробці складних запитів користувачів. [1]

Сьогодні Семантичний Веб представлений наступними технологіями:

- глобальна схема імен – URI (Uniform Resource Identifier)

URI – це уніфікований ідентифікатор ресурсів у Вебі. Будь-який елемент контенту, що має URI вважається “присутнім у Вебі”. Кожний об’єкт даних та кожна схема/модель даних у Семантичному Вебі повинна мати власний URI.

- стандартний синтаксис для опису даних – RDF (Resource Description Framework)

RDF (діалект XML) – це специфікація, яка визначає модель для представлення світу і синтаксис для зміни цієї моделі. RDF забезпечує стандартизований підхід до опису різноманітних Веб-ресурсів (Веб-сторінок, зображень, аудіо- та відеофайлів тощо) і є базовою мовою розмітки для Семантичного Вебу, яка визначає напрямлений граф відношень. Такий граф представлений трійкою “об’єкт-атрибут-значення” (об’єкт А має атрибут В зі значенням С). Альтернативою RDF XML є Notation3.

- стандартний засіб опису властивостей даних – RDF Schema

RDF Schema – це семантичне розширення RDF, яке забезпечує механізм для опису груп зв'язаних ресурсів та зв'язків всередині цих груп.

- стандартний засіб опису зв'язків між об'єктами даних – онтології, описані за допомогою OWL Web Ontology Language

Синтаксична взаємодія полягає у коректному синтаксичному аналізі даних і вимагає побудови відповідностей між термінами, яка в свою чергу вимагає аналізу контенту.[2]

Аналіз контенту вимагає формальних явно заданих специфікацій моделей доменів, які визначають використані терміни та зв'язки між ними. Такі моделі доменів називають онтологіями (ontologies). Онтології визначають моделі даних у термінах класів, підкласів та властивостей.

Саме для вираження онтологій і була створена мова OWL (Ontology Working Language). OWL має більше засобів для опису класів і властивостей, ніж RDF та RDF Schema, зокрема можливість опису зв'язків між класами, кількості елементів тощо.[2]

OWL призначений для використання додатками, метою яких буде опрацювання Веб-контенту, і сприяє наданню Веб-контенту ширших можливостей бути інтерпретованим цими додатками.

- стандартна мова запитів до даних у форматі RDF – SPARQL (рекурсивно: SPARQL Protocol and RDF Query Language)

Мова запитів SPARQL дозволяє видобувати інформацію з графів RDF. Вона надає засоби для:

- видобування інформації у формі: URI, порожніх вершин графу, типізованих або нетипізованих літералів;
- видобування підграфів RDF;
- створення нових графів RDF, заснованих на інформації з графів, до яких здійснювалися запити.

Як мова запитів, SPARQL дозволяє працювати як з локальними, так і віддаленими даними. [2]

1.6 Перспективи використання семантичних веб-додатків

Є можливість, що завдяки семантичним веб-додаткам Internet зможе вийти з наміченої кризи, пов'язаної з "проблемою розмірності". З'явилася надія, що комп'ютери зможуть обробляти дані відповідно до їх змісту, слідуючи за гіпер-посиланнями, що ведуть до визначень ключових термінів і правил логічних висновків. Отримана в результаті інфраструктура дасть відправну точку для розробки автоматизованих Web-сервісів, інтелектуальних агентів, адже сама ідея Семантичного Web заснована на прагненні "навчити" комп'ютерні програми, Web-служби і роботів пошукових систем і агентів "осмислено" оперувати тією інформацією, для якої останні були створені.[3]

Семантичний Web обіцяє цілком відчутні переваги, додаткові сервіси. Навігація в Мережі стане більш осмисленою, а пошук - більш точним. Самі користувачі зможуть створювати сторінки Семантичного Web, давати власні визначення і вводити нові правила виведення, використовуючи стандартне для цієї мережі програмне забезпечення.

Семантичний Web - це не якась окрема мережа, а розширення і еволюція вже існуючої, але при цьому інформація має точно визначений сенс, що дозволяє людині і програмам успішно взаємодіяти. Сьогодні відбувається активна інтеграція нових елементів Семантичної Мережі в структуру традиційного Web. Семантичний Web вже цілком готовий до широкого впровадження в корпоративному секторі, він переріс межі суто дослідницького проекту, всі його основні технології стають стандартами, а великі учасники ринку високих технологій впроваджують їх у прикладні програми корпоративного рівня.

В даний час на Семантичний Web працює безліч наукових підрозділів по всьому світу, удосконалюючи і розробляючи нові протоколи, технології, середовища програмування, агенти, мови, інтерфейси, методи розподіленого пошуку знань. Прогнозується, що працездатна глобальна версія Семантичної мережі з'явиться вже в цьому десятилітті. Про реальність цього прогнозу свідчить публікація та затвердження WWW-консорціумом у лютому 2004 року фінальних версій двох основних специфікацій Семантичного Web. Це переглянута версія RDF (до неї додані опису тестів, що дозволяють додаткам на різних мовах програмування розуміти один одного, а також засоби стикування RDF і XML) і OWL.

Ось список лише деяких з безлічі подій, що відбулися за рік, що минув з "дня народження" Семантичного Web:

- 10 лютого 2004 - Консорціум W3C публікує ключові компоненти глобального проекту Семантичного Web: рекомендації RDF і Web Ontology Language (OWL). Багато експертів вважають цю дату офіційним днем народження Семантичної Мережі.
- серпня 2004 - Робоча група Semantic Web Best Practices and Deployment видала першу версію специфікації значень OWL, що складається з двох частин: "набір значень" і "розподіл значень". У специфікації представлений великий набір різних елементів мови OWL, причому автори пропонують усім зацікавленим особам і організаціям включатися в процес доопрацювання і поповнення цього набору.
- 22 листопада 2004 вийшов у світ ще один комплект документації, присвяченій технологіям Семантичного Web. Рекомендації OWL-S: Semantic Markup for Web Services присвячені створенню онтологій і застосуванню мови OWL як елемента Web-служб для автоматизації процесів компоновки, дослідження, активізації та моніторингу

ресурсів Мережі. Уродженець Великобританії, винахідник World Wide Web Тім Бернерс-Лі (Tim Berners-Lee), який зараз живе в США, в 2004 році за це свій винахід отримав від королеви Великобританії Єлизавети II лицарський титул. У квітні 2004 року за цей же винахід сер Тім Бернерс-Лі став першим лауреатом нової премії Millennium Technology Prize, грошовий еквівалент якої становить 1 млн євро. Крім того, він визнаний "Найвидатнішим британцем 2004 року".[3]

Тім Бернерс-Лі закінчив Королівський коледж Оксфордського Університету, після чого працював в CERN (Європейський Центр Досліджень елементарних частинок). Там в 1980 р він написав програму Enquire, призначену для зберігання інформації, в якій використовувалися випадково встановлювані зв'язки і яка стала предтечею WWW. Пізніше в 1989 році він ввів термін Web і створив мову гіпертекстової розмітки HTML. Потім в 1990 році з'явилися перші HTTP-сервер і перший Web-браузер. Всесвітня павутина WWW, як система доступу до інформації, почала працювати в 1991 р. У 1994 р Бернерс-Лі, в той час співробітник Массачусетського технологічного інституту (США), заснував і очолив некомерційну організацію консорціум World Wide Web Consortium (W3C), яка займається технічними проблемами розвитку і функціонування Web. В даний час Тім Бернерс-Лі очолює розробку концепції Семантичного Web.[3]

1.7 Композитні семантичні веб-додатки, задачі композиції веб-сервісів у семантичному веб просторі.

Композиція Веб-сервісів викликає значний інтерес у контексті проблематики семантичної Веб-мережі. Її цілі полягають у тому, щоб зробити існуючий Веб більш ефективним і забезпечити гнучкий підхід до управління всіма типами процесів. Очікувані програмні додатки, побудовані на основі

композиції сервісів, зроблять більш перспективними такі напрямки як Веб-комерція, електронний уряд, електронна наука та інші.[3]

На сьогодні композиція сервісів виконується, як правило, вручну, в процесі розробки додатків. Це вимагає багато часу і програмування на досить низькому рівні. На даний час стає все більш досліджуваною модель композиції Веб-сервісів, яка базується на онтологіях. Підхід до генерації композитного сервісу заснований на декларативних описах. Формальною основою для композиції служать композиційні правила. Ці правила поєднують синтаксичні та семантичні характеристики сервісів для визначення того, чи є сервіси композитними.

Концепція Семантичного Веб спрямована на розширення існуючого Веб в частині визначення та обробки змісту інформації. Кінцева мета Семантичної мережі Інтернет полягає в тому, щоб перетворити Веб в середу, через яку дані можуть бути розділені, зрозумілі і оброблені автоматизованими або автоматичними засобами, тобто машинами.[3]

Розвиток технологій Семантичної Веб-мережі є пріоритетним завданням багатьох дослідницьких проектів. Одним з ключових проблемних понять, в дослідженнях є поняття Веб-сервісу, який являється програмною системою, представленою безліччю пов'язаних функцій, до яких можна звернутися програмно через мережу. Веб-сервіси призначені для вирішення проблем інтероперабельності і повторного використання програм при взаємодії машина-машина в мережі. Поняття сервісу поступово впроваджується в бізнес, державну діяльність, науку та інші галузі. Прикладами Веб-сервісів у бізнесі є електронна торгівля акціями, перевірка кредитів, електронні платежі.

Веб-сервіси в державних структурах забезпечують функції різних соціальних фондів.

Дуже поширені стандарти на основі XML, включаючи: WSDL, SAWSDL, SOAP, UDDI, WSMO і MSSLI, ініціюють інтенсивні дослідження Веб-сервісів в промисловості та академічних проектах. Одним з найважливіших призначень Семантичної мережі Інтернет є використання її як носія сервісів. Ця нова

модель дала б можливість компаніям значно зменшити витрати на розгортання нових рішень бізнесу та відкриття нових можливостей.[3]

Ідентифікують два типи сервісів: прості і композитні. Простий сервіс представляє програмну систему, створену на основі Інтернету, яка не використовує інші Веб-сервіси для виконання запиту користувача. Складний або композитний сервіс визначається тим, що він використовує сукупність функцій простих або композитних сервісів для виконання своїх завдань. Прикладом композитного сервісу може бути автомобільний брокер, який отримує дані для виконання робіт від агента по продажу автомобілів, фінансування та страхового обслуговування для забезпечення сервісу продажу.

Композиція сервісів стала центральним завданням у дослідженнях по Веб-сервісів. У цій області запропоновано декілька нових підходів. Як правило, всі вони вимагають детального програмування, що робить процес композиції сервісів залежним від програміста-розробника сервісів.

Розробники композиції повинні визначити, які дії повинні бути виконані композитним сервісом, і не цікавитися, які сервіси необхідно залучити до виконання і як залучені сервіси повинні взаємодіяти. Процес композиції сервісів включає такі операції: вибір сервісів, вбудовування операцій, визначення повідомлень розмови та інші, які повинні бути прозорими для користувача. Деталізовані описи композитного сервісу повинні бути згенеровані з специфікації композиції.[3]

Семантика є важливою і основною складовою для забезпечення композиції сервісів, хоча б перевірки того, що обрані для композиції сервіси були коректними і відповідали вимогам композиції. Такі засоби можуть бути синтаксичними (наприклад, перевірка числа параметрів, включених до повідомлення, яке надсилається або отримується сервісом). Вони також можуть бути семантичними (наприклад, склад бізнес-функцій, які пропонуються сервісною операцією або визначення відповідності предметної області). Для визначення семантичних характеристик Веб-сервісів використовується поняття онтології, яка визначається як концептуалізація, що бере за основу семантичну

близькість термінів у певній предметній області. Онтологія — ключове поняття для формального визначення семантики і семантичного управління процесами. Онтології відіграють центральну роль в Семантичній мережі Інтернет, розширюючи синтаксичну інтероперабельність сервісів до їх семантичної інтероперабельності.

Процес створення Веб-сервісу вимагає його опису для того, щоб інші сервіси могли зрозуміти його можливості і дізнатися, як взаємодіяти з ним. Мова WSDL (мова опису Веб-сервісів) призначена для опису операційних особливостей Веб-сервісів. WSDL стандартизується в межах консорціуму W3C. Головні лідери промисловості підтримують та беруть участь у розвитку WSDL. Однак WSDL майже не забезпечує семантичний опис Веб-сервісів. Вона головним чином включає конструкції, які описують Веб-сервіс з синтаксичної точки зору. Щоб відповідати визначенню Семантичного Веб-сервісу, використовується підхід, заснований на розширенні WSDL семантичними характеристиками. Цей підхід покладений в основу вибору і композиції Веб-сервісів з використанням стандартів WSDL 2.0, WSDL 4j і SAWSDL. Для моделювання онтології скористуємося моделлю, яка базується на орієнтованих графах, вузли якого представляють концепти онтології. Безліч вузлів графа поділяється на «Заповнених» і «незаповнених». «Незаповнені» вузли посилаються на концепт, які визначені мовою WSDL (Наприклад, імена, зв'язку, входи). «Заповнені» вузли посилаються на описи, що отримуються за допомогою розширення WSDL семантичними характеристиками. Дуги представляють відношення між поняттями онтології. Їх маркують кількістю елементів відповідних відношень. Наприклад, дуга «сервіс> операція» означає те, що сервіс має одну або більше операцій. Дуга «операція>> Вхід» означає те, що операція має більше одного вхідного повідомлення. Визначення Веб-сервісу полягає в визначенні кожного концепту відповідної онтології.[3]

1.8 Висновки

Отже, композитні семантичні веб-додатки це новий крок у розвитку використання інтернету. ‘Розумні’ веб-сервіси, що можуть як завгодно поєднуватися, використовувати дані і функціонал один одного зроблять мережу більш зручною, ефективнішою, гнучкою у використанні.

2 БАЗИ ЗНАНЬ. ОНТОЛОГІЇ. СЕМАНТИЧНІ ЗВ'ЯЗКИ У НАУКОВИХ ДОСЛІДЖЕННЯХ

2.1 Семантичний Web

Отже, Семантичний Web можна представити як симбіоз двох напрямків, перший з яких охоплює мови представлення даних. На сьогоднішній день основними такими мовами є Розширюваний Мова Розмітки XML (eXtensible Markup Language) і Засоби Опису Ресурсів RDF (Resource Description Framework). Існує також ряд інших форматів, однак XML і RDF надають більше можливостей, тому вони володіють статусом рекомендацій W3C.

Другий, концептуальний напрямок несе в собі теоретичне уявлення про моделі предметних областей. Такі моделі предметних областей в термінології Семантичного Web називаються онтологіями. 10 лютого 2004 консорціумом W3C була затверджена й опублікована специфікація мови мережевих онтологій OWL (Web Ontology Language).

Таким чином, дві гілки Семантичного Web використовують три ключових мови (відповідно, технологій):

- Специфікація XML, що дозволяє визначити синтаксис і структуру документів;
- Механізм опису ресурсів RDF, що забезпечує модель кодування для значень, визначених в онтології.
- Мова онтологій OWL, що дозволяє визначати поняття і відносини між ними. Семантичний Web використовує також і інші мови, технології та концепції, зокрема, універсальні ідентифікатори ресурсів, цифрові підписи, системи логічного висновку і т.п.

2.2 Список основних діючих рекомендацій W3C, пов'язаних з Семантичним Web

- XML (www.w3c.org/XML) забезпечує синтаксис для структурованих документів, але не накладає ніяких семантичних обмежень на зміст цих документів
- XML Schema (www.w3c.org/XML/Schema) визначає структуру документів XML, а також доповнює XML конкретними типами даних.
- RDF (www.w3c.org/TR/2002/WD-rdf-concepts-20021108) дозволяє описати модель даних для ресурсів і відносини між ними, забезпечує просту семантику для цієї моделі даних, представляючи їх в синтаксисі XML.
- RDF Schema (www.w3c.org/TR/2002/WD-rdf-schema-20021112) надає засоби для опису властивостей і класів RDF-ресурсів, а також семантику для ієрархій-узагальнень таких властивостей і класів.
- OWL (<http://www.w3.org/TR/owl-features/>) розширені можливості опису властивостей і класів.

Якщо говорити про логічні рівні, на яких базується технологія Семантичного Web, то самий нижній рівень - це Universal Resource Identifier (URI), уніфікований ідентифікатор, що визначає спосіб запису адреси якого-небудь ресурсу. Семантичний Web, іменуючи всяке поняття просто за допомогою URI-ідентифікатора, дає можливість кожному висловлювати ті поняття, якими він користується. Типовими прикладами URI-ідентифікаторів є URL-адреси, однак URI-ідентифікатор задаючи або посилаючись на деякий ресурс, не обов'язково при цьому вказує на його місцезнаходження в Internet.

Наступний рівень - мова XML як базова форма розмітки та засоби, призначені для визначення та опису класів XML-документів (DTD, XML- схеми). Окремий рівень в концепції Семантичного Web орієнтований на роботу з цифровим підписом, який необхідний, щоб клієнти могли визначати ступінь достовірності даних.

Крім того на базі XML розгортаються засоби опису ресурсів RDF і RDF-схеми, що пояснюють, як зістиковувати XML-дані в мережі і будувати каталоги і словники понять. RDF дозволяє виконувати пошук необхідних понять в Семантичному Web.

І нарешті, мова мережевих онтологій OWL призначений для опису класів і відносин між ними, які притаманні як для мережевих документів, так і додатків. OWL забезпечує більш повну автоматичну обробку мережевого контенту, ніж та, яку підтримують XML і RDF, надаючи поряд з формальною семантикою додаткову семантичну підтримку. При цьому самі онтології утворюють систему, що складається з наборів понять і тверджень про ці поняття, на основі яких можна будувати класи, об'єкти і відносини. Окрема онтологія визначає семантику конкретної предметної області і сприяє встановленню зв'язків між значеннями її елементів.

2.3 Структура семантичного Web

Першим спеціалізованим інструментом Семантичного Web, виявилась мова XML (eXtensible Markup Language), віднесена в підсумковому рейтингу компанії IDG за 2001 рік до числа головних технологічних досягнень, здатних змінити в найближчі роки весь комп'ютерний світ.

Як мова XML являє собою досить універсальну синтаксичну основу, що забезпечує як представлення даних, так і задання відносин між ними. Крім того, XML являє собою відкриту семантично орієнтовану технологію, яка надає широкі можливості роботи з метаданими.

Вихідна версія XML, розроблена в консорціумі W3C під керівництвом Джона Босака, була опублікована в лютому 1998 р і з тих пір розвинулася до рівня метамови, на базі якого визначаються сотні нових предметно-орієнтованих мов (наприклад, MathML, XLink, SMIL, XSL та ін.)

На відміну від HTML, що створювався для гіпертекстових документів з фіксованою структурою і складом тегів, XML призначений для розмітки

документів довільної структури. XML-документи можуть містити збалансовані дерева вкладених відкриваючих і закриваючих тегів, кожен з яких може включати в себе кілька пар "атрибут - значення". Однак XML як мова сама по собі нічого не говорить про семантику понять, що складають його синтаксичні структури.

Мова XML дозволяє створювати свої власні теги - приховані мітки, якими можна постачати Web-сторінки або розділи тексту на сторінках для опису їх інформаційного наповнення. Наприклад, тег <ідентифікатор> зміст </ідентифікатор>, який користувач потім повинен наповнити актуальною інформацією, наприклад: <DateOfBirth> 30-09-1987 </ DateOfBirth>. Такий підхід застосуємо практично для всіх областей, наприклад, бібліотекарі могли б визначити власні теги для обміну даними про книги, зокрема, <ISBN>, <Author> (<Автор>), <Price> (<Ціна>) і т.д ., а потім зберегти ці визначення в деяких схемах. Після чого XML-орієнтовані бібліографічні програми зможуть автоматично обробляти такі документи. При цьому XML, насправді, "не розуміє" і не "роз'яснює" змісту документів. Комп'ютерні програми можуть використовувати XML-теги, але при цьому в програмах має бути на алгоритмічному рівні закладено "розуміння" їх призначення.

Оскільки в XML не існує фіксованого словника тегів, то вони можуть визначатися незалежно для кожної програми. У XML це було спочатку передбачено за допомогою визначення типу документа DTD (Document Type Definitions), що накладає обмеження на використовувані теги і задає граматику, яка вказує допустимі комбінації і вкладення імен тегів, імен атрибутів і т.д. Сьогодні практично в кожній галузі знань є свій, постійно розширюючийся список DTD. У XML застосовується два види вказівки на DTD:

- Декларації внутрішніх підмножин DTD-визначень, які розміщені посередньо в XML-документ. При цьому команду визначення DTD вставляють у квадратні дужки, наприклад: <!DOCTYPE rootElement [declarations]>;

- Посилання на зовнішні DTD-визначення, наприклад, `<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">`

Вказівка PUBLIC в другому випадку свідчить про те, що DTD є загальнодоступним описом, зокрема стандартом мови WML.

Мова DTD дозволяє визначити логічну структуру документа, тобто:

- Вказувати порядок проходження елементів;
- Визначати вкладеність елементів;
- Встановлювати кількість можливих елементів;
- Встановлювати типи атрибутів;
- Визначати сутність і нотації.

Разом з тим, мові DTD притаманні два серйозні недоліки - обмеженість опису типів даних і синтаксис, відмінний від XML. Тому в даний час консорціум W3C наполегливо рекомендує замінювати використання DTD новим стандартом - XML-схем (XML Schema), який був затверджений в 2001 році (<http://www.w3.org/TR/xmlschema-formal/>).

Провідні виробники програмного забезпечення в усьому світі прийняли концепцію XML-схем і впровадили її в своїх продуктах. Так корпорація Microsoft надала в доступ на умовах безкоштовного ліцензування схем в Microsoft Office 2003 XML розробникам. В даний час за адресою www.microsoft.com/office/xml/default.mspx доступні XML-схеми:

- SpreadsheetML для Microsoft Office Excel 2003;
- FormTemplate Schemas для Microsoft Office InfoPath 2003;
- WordprocessingML для Microsoft Office Word 2003;
- DataDiagramingML для Microsoft Office Visio 2003.

На думку представників корпорації, доступність схем значно полегшить реалізацію підтримки можливості обміну даними між програмами, що розробляються і додатками офісного пакету.

Ще одна безумовна перевага XML полягає у використанні ним сучасного стандарту кодування символів Unicode, який дозволяє комбінувати тексти,

написані на всіх основних мовах світу (у тому числі, є підтримка кирилиці), в одному документі. Тим самим XML дає можливість з легкістю обмінюватися інформацією поза національними кордонами.

Оскільки XML є технологічним стандартом, можна говорити і про хорошу подальшу перспективу його використання і в якості стандарту системи управління знаннями.

Сьогодні у всьому світі XML вступив у фазу широкомасштабного впровадження. За допомогою цієї технології здійснюється управління інформацією, класифікаційними схемами, індексування.

2.4 Засоби опису ресурсів RDF

Важлива компонента Семантичного Web - методологічна модель RDF, призначення якої полягає в описі відносин між мережевими ресурсами та інформацією. Адже самі по собі теги XML відірвані від їх змістового наповнення. Тому паралельно з XML було розпочато розробку стандарту RDF як мови формального опису змісту мережеских ресурсів, який згідно архітектурі Семантичного Web, являє собою сполучну ланку між XML-документами і засобами, що забезпечують пошук і навігацію на основі логічних тверджень. RDF являє собою технологію для вираження сенсу термінів і понять у вигляді, доступному для обробки програмами. Ця технологія призначена для стандартизації визначень і використання метаданих, що описують Web-ресурси, а також для представлення самих даних, що містяться в цих ресурсах.

Принцип побудови відносин між мережевими ресурсами в специфікації RDF передбачає наявність трьох компонент - об'єкта, атрибута і значення (аналогічних класичною схемою "підмет - присудок - доповнення"). Кожному елементу даного ланцюжка (триплету) присвоюється ідентифікатор (URI), за допомогою якого при вказівці однієї з ланок можна автоматично відновити весь ланцюжок в цілому. З триплетів мови RDF можуть формуватися мережі з взаємопов'язаних об'єктів. Оскільки RDF використовує URI-ідентифікатори для

кодування інформації в документі, це забезпечує те, що кожне поняття буде прив'язано до єдиного визначення, яке можна знайти в Мережі.

Базовий будівельний блок в RDF - трійка "об'єкт - атрибут - значення" часто записують у вигляді A (O, V), де O - об'єкт, A - атрибут зі значенням V. RDF дозволяє міняти місцями об'єкти і значення.

Спочатку в RDF використовувався синтаксис мови XML і URI-ідентифікатори для вказівки об'єктів, понять, властивостей і відносин. Однак існують і інші форми RDF-описів, наприклад, у вигляді наборів трійок:

```
hasName('Http://dwl.visti.net', "Dmitry Lande")
authorOf('Http://dwl.visti.net/', 'ISBN5845907640')
hasPrice('ISBN5845907640', "$ 8").
```

Крім того, RDF допускає форму подання, в якій будь-який вираз RDF в триплеті може бути об'єктом або значенням, тобто це дозволяє вказати, що даний об'єкт має певний тип, наприклад, що "ISBN5845907640" - це rdf: type book, за рахунок вказівки на визначення book у схемі RDF:

```
<Rdf: Description about = "www.book.net/ISBN5845907640">
<Rdf: type rdf: resource = "http://description.org/schema/#book">
</ Rdf: Description>
```

Специфікації RDF забезпечують підтримку тегів, що дозволяють визначати практично будь-які поняття (наприклад, в електронній комерції можна користуватися такими тегами, як "Price" і "Invoice", яким приписати значення ціни та рахунку, відповідно).

Інформаційні ресурси описуються за допомогою сукупності термінів. Семантика, сенс самих термінів і словників термінів, фіксується за допомогою глобальних універсальних імен URI. У RDF передбачений словник термінів, званий RDF-схемою, що визначає, які терміни можуть бути використані в RDF-твердженнях про властивості ресурсів, являє ієрархію понять предметної області, описує характеристики кожного з термінів.

У структурі Семантичного Web передбачені й більш ефективні спеціальні засоби - онтології, що представляють собою універсальні бази знань. Ці бази знань містять відомості для ототожнення нових понять з уже відомими, для визначення приналежності використовуваних термінів до тієї чи іншої предметної області і, врешті-решт, для приведення будь-яких понять до вигляду, придатного для сприйняття програмами - електронними агентами.

2.5 Онтології

У філософії онтологією називають теорію про природу буття і видах сутностей. Онтологічний рівень формалізує накопичені знання, визначаючи і об'єднуючи термінологію різних предметних областей.

Онтології отримали досить широке поширення в задачах представлення знань та інженерії знань, семантичної інтеграції інформаційних ресурсів, інформаційного пошуку і т.д. У науці про "штучний інтелект" онтологія - це "специфікація концептуалізації предметної області", або спрощено, документ або файл, що формально задає відносини між термінами. Це свого роду словник понять предметної області і сукупність явно виражених припущень щодо змісту цих понять.

Найчастіше онтологія представляється як ієрархія понять, пов'язаних відносинами деяких певних видів. Такі визначення онтологій використовуються в різних класифікаціях. Розвинені онтології формалізуються засобами мов логіки і допускають можливості логічного висновку.

У найпростішому випадку онтології можна використовувати для підвищення точності пошуку в Internet - пошукова система буде видавати тільки такі сайти, де згадується в точності шукане поняття, а не довільні сторінки, в тексті яких зустрілося задане ключове слово.

Загальновідомо, що в різних предметних областях одні й ті ж поняття можуть бути представлені різними термінами. Механізм онтологій в цих випадках дозволяє формувати осмислені ієрархічні взаємозв'язки між

об'єктами, узагальнювати і спільно використовувати глобальні відомості, тобто реалізувати нечіткий пошук, здатний знаходити навіть такі необхідні користувачу ресурси, в яких не буде жодного слова з вихідного запиту.

Передбачається, що "інтелектуальні" програми будуть використовувати онтології, щоб одержувати в результаті пошуку інформацію з пов'язаною з нею структурою знань і правилами виведення. Механізми пошуку можуть застосовувати онтології і для вибірки сторінок з синтаксично різними, але семантично однаковими словами. Онтології також можуть використовуватися для організації обміну даними та інтеграції програм.

Така інтелектуальна програма, що інтерпретує онтології, зможе вивести, наприклад, що якщо Університет Корнелла знаходиться в м. Ітака, який розташований в штаті Нью-Йорк, який, у свою чергу, є частина США, то адресу цього університету слід писати в американському форматі.

Розробка мови опису структурованих онтологій OWL стало останнім часом одним з найбільш важливих ланок робіт з Семантичний Web, що проводяться консорціумом W3C. В кінці 2001 року для цієї мети в складі W3C була заснована спеціальна робоча група - Web Ontology Working Group. 10 лютого 2004 WWW- Консорціум присвоїв мові OWL статус рекомендованої до реалізації технології. В рамках OWL онтологія - це сукупність тверджень, які задають відносини між поняттями і визначають логічні правила для міркувань про них. Комп'ютери можуть "розуміти" сенс семантичних даних на Web-сторінках, слідуючи за гіперпосиланнями, ведучим на онтологічні ресурси. Онтологія може включати описи класів, властивостей і їх приклади (індивіди).

Формальна семантика OWL описує, як отримати логічні висновки на основі онтологій, тобто отримати факти, які не представлені буквально, а випливають з семантики онтології. Ці висновки можуть базуватися на аналізі одного документа або безлічі документів, розподілених в Мережі. Останнє забезпечується можливістю онтологій бути пов'язаними, включаючи прямий імпорт інформації з інших онтологій. Щоб написати онтологію, яка може

однозначно інтерпретуватися і використовуватися програмними агентами, задіюються синтаксис та формальна семантика OWL.

На практиці створення онтологій починається з ієрархії класів понять, що складають предметну область. Фундаментальним таксономічним конструктором для класів є `rdfs: subClassOf`. Він пов'язує більш приватний клас з більш загальним класом. Якщо X - підклас Y , то кожен представник X - також представник Y . Ставлення `rdfs: subClassOf` є транзитивним. Якщо X - підклас Y , і Y - підклас Z , то X - підклас Z . Нижче наведені описи класів і підкласів, запозичені з рекомендацій W3C і пов'язані з предметною областю - виноробством.

```
<Owl: Class rdf: ID = "Напій">
<Rdfs: subClassOf rdf: resource = "# ПродуктХарчування" />
...
</ Owl: Class>
```

У даному прикладі Напій визначено як підклас класу ПродуктХарчування. Нижче дано спрощене визначення для класу Вино, яке є Напоєм:

```
<Owl: Class rdf: ID = "Вино">
<Rdfs: subClassOf rdf: resource = "Напій" />
<Rdfs: label xml: lang = "en"> wine </ rdfs: label>
<Rdfs: label xml: lang = "ru"> вино </ rdfs: label>
<Rdfs: label xml: lang = "fr"> vin </ rdfs: label>
...
</ Owl: Class>
```

Конкретний об'єкт в OWL Прийнято називати індивідом. У прикладі ВиноградКабернеСовіньон - це індивід, бо позначає один конкретний сорт винограду:

```
<Owl: Class rdf: ID = "ВиннийВиноград">
<Rdfs: subClassOf rdf: resource = "& food; Виноград" />
</ Owl: Class>

<ВиннийВиноград rdf: ID = "ВиноградКабернеСовіньон" />
```

В OWL існує безліч способів обмеження відносин. Наприклад, можна визначити домен і діапазон, - так властивість може бути визначена як підвластивість (спеціалізація) вже існуючої властивості.

```
<Owl: ObjectProperty rdf: ID = "зробленоЗВинограду">
<Rdfs: domain rdf: resource = "# Вино" />
<Rdfs: range rdf: resource = "# Виноград" />
</ Owl: ObjectProperty>
```

У даному прикладі властивість зробленоЗВинограду має домен Вино і діапазон Виноград. Таким чином, це пов'язує представників класу Вино з представниками класу Виноград. Множинні домени означають, що доменом властивості служить перетин зазначених класів.

За словами Тіма Бернерса-Лі в прес-релізі W3C, RDF і OWL - це серйозний крок і вельми потужна база для додатків Семантичного Web. Затвердження цих стандартів як рекомендованих W3-консорціумом було дуже вчасно. Сьогодні відкривається нова фаза Internet як інформаційного простору. Ця фаза почалася з того моменту, коли проект Семантичний Web почав свою роботу.

2.6 Знання

Проект Семантичний Web припускає створення системи з елементами "штучного інтелекту", яка б дозволила спеціальним програмам якісно шукати в Internet необхідну інформацію, а також обмінюватися інформацією один з одним. При цьому саме мова онтологій OWL виступила вирішальною компонентою інтелектуалізації, базисом для побудови семантичних мереж. Необхідно відзначити, що вперше теорія семантичних мереж народилася в середині минулого століття і була орієнтована на завдання штучного інтелекту, зокрема, машинного перекладу. Знання в теорії семантичних мереж представлялися у вигляді вузлів, з'єднаних дугами, кожна з яких визначала тип ставлення. Семантичний Web по суті є реалізацією ідеї штучного інтелекту, однак цей термін не дуже популярний зважаючи на велику кількість невдалих

проектів у цій галузі, тому поняття "семантична мережа" сьогодні викликає настороженість. Разом з тим, Web онтології по суті являють собою справжню базу знань, одну з концептуальних основ штучного інтелекту.

Уявленням знань в Семантичному Web притаманні універсальні виразні можливості, синтаксична і семантична інтероперабельність. Дійсно, мови Семантичного Web дозволяють представляти будь-який вид даних, створюються численні синтаксичні аналізатори та інтерфейси прикладних програм необхідні для маніпулювання даними. Семантична інтероперабельність реалізується, наприклад, в онтологіях шляхом встановлення відповідності між використовуваними термінами.

2.7 Агенти

За словами Тіма Бернерс-Лі, Семантичний Web запрацює на повну силу тоді, коли люди створять безліч програм, які, знайомлячись з вмістом Мережі з різних джерел, зможуть обробляти отриману інформацію і обмінюватися результатами з іншими програмами. Семантичний Web повинен створити середовище, в якому програмні агенти, переходячи зі сторінки на сторінку, зможуть без особливих зусиль виконувати витончені запити користувачів.

Під агентами в даному випадку розуміються програми, що працюють без безпосереднього управління з боку людини для досягнення поставлених перед нею цілей. Зазвичай агенти збирають, фільтрують і обробляють інформацію, знайдену в Мережі, іноді шляхом взаємодії з іншими агентами.

Основні принципи функціонування автономних агентів полягають в наступному:

- Агент не має повної інформації, необхідної для вирішення поставленого завдання;
- Оброблювані дані розподілені в мережі;
- Обчислення виконуються агентами асинхронно;

- Взаємодія агентів один з одним і з людиною відбувається на високому семантичному рівні;

- Відсутній глобальний контроль за діяльністю всієї системи агентів.

У Семантичному Web агенту зовсім не обов'язково володіти штучним інтелектом, а просто реалізувати чіткий алгоритм при роботі з базою знань онтологій. Передбачається, що вся семантика буде закладена в Web-сторінку автором, який з цією метою буде використовувати стандартне програмне забезпечення для написання сторінок Семантичному Web.

Передбачається, що ефективність програмних агентів в Семантичному Web буде рости в міру збільшення кількості доступного їм Web-контенту та автоматизованих сервісів (включаючи інших агентів). З іншого боку, навіть ті агенти, які були створені спеціально для спільної роботи, можуть передавати інформацію між собою.

У мові OWL є ряд можливостей, що вже існували в мовах подібного призначення, наприклад, в DARPA Agent Markup Language (DAML, мова спілкування автономних агентів), запропонований військовим науковим агентством DARPA для автоматизації управління складними об'єктами.

Важливим аспектом функціонування агентів в Семантичному Web є змога обміну висновками і міркуваннями, проведеними з використанням інформації, що утримується в онтологіях та правил виводу.

Життєво важливою функціональністю агентів в Семантичному Web будуть і цифрові підписи, які будуть використовуватися для підтвердження того, що інформація надана тим чи іншим джерелом.

2.8 Виявлення сервісів

В даний час існує безліч автоматизованих Web-сервісів, побудованих без урахування можливостей Семантичного Web. При цьому автоматично виявити в мережі подібну програму далеко не просто. Гарантоване виявлення сервісів стане можливим лише після того, як широко розповсюдиться уніфікована мова,

що дозволяє описувати сервіси, з тим щоб відповідні програмні агенти могли "розуміти", що дозволяє робити даний сервіс і яким чином їм користуватися.

Технологія виявлення Web-сервісів повинна знайти своїх користувачів, наприклад, у сфері малого бізнесу, де стане набагато простіше налагоджувати проведення платіжних операцій в галузі електронної комерції, що мають велику ступінь захисту.

Поза Семантичного Web сьогодні доступні деякі схеми виявлення сервісів. Так, наприклад, Universal Plug and Play фірми Microsoft вирішує проблему підключення різних типів пристроїв, а система Jini компанії Sun Microsystems призначена для з'єднання сервісів. У Семантичному Web виявлення сервісів має бути більш гнучким - Web-сервіси та агенти можуть досягти "взаєморозуміння" шляхом обміну онтологіями, що містять необхідні для спілкування термінологічні словники. Більше того, агенти зможуть навіть, знаходячи нові онтології, удосконалювати свої алгоритми. За допомогою семантичних механізмів також стане простіше виявляти сервіси, які лише частково відповідають критерію пошуку.

Ідеологія систем "інтелектуальних агентів" цілком укладається в існуючі сьогодні технології CORBA, EJB, .NET та ін. Агенти створюються насамперед на Java як найбільш поширеній мові мережевого програмування, до того ж незалежним від платформи.

Найбільший з відомих відкритих проєктів з розвитку Java-агентів - Cougaar (cougaar.org/projects/cougaar/), дозволяє створювати масштабні розподілені інтелектуальні комплекси планування дій, що відрізняються високою стійкістю при роботі у швидко мінливих мережевих умовах.

У 2003 р DARPA відкрило п'ятирічну програму-Perceptive Assistant that Learns (PAL) з бюджетом 29 млн. дол. У відповідності з цією програмою передбачається створення інтелектуальних помічників, здатних динамічно навчатися при спілкуванні з людиною і накопичувати знання про конкретну предметну область. Володіючи навичками самонастроювання і високої адаптації, PAL-агенти навчаються управлінню доступними ресурсами і

розстановці пріоритетів при виконанні безлічі завдань, а також взаємодіють один з одним.

Ще одна відома робота DARPA в цій сфері - CoABS (<http://coabs.globalinfotek.com>), що об'єднує кластери користувачів (людей і програми) зі спільними інтересами. Вона заснована на проекті Массачусетського технологічного інституту Yenta (www.media.mit.edu/people/foner/Yenta/overview.html), орієнтованого на використання агентів при вирішенні керуючих, інформаційних та логічних задач.

Серед вільно доступних і постійно розвиваються систем для створення Java-агентів і систем онтологій можна назвати й такі:

Protigi-2000 (<http://protege.stanford.edu>) - проект Станфордського університету, що нараховує майже 18 тис. Користувачів.

JTP (Java Theorem Prover, www.ksl.stanford.edu/software/JTP/) - об'єктно орієнтована система логічного висновку;

AGLET (www.trl.ibm.com/aglets/) - проект наукового підрозділу корпорації IBM по створенню аглетів, або мобільних агентів, що самостійно переміщаються по мережі.

OAA (Open Agent Architecture, www.openagent.com) - створювана компанією SRI International архітектура програмування, що забезпечує динамічну взаємодію програм в гетерогенних мережах для вирішення одного завдання.

Семантичний Web не скасовує корисності Web-сервісів, а доповнює їх. Однак, у той час, як завдання більшості сучасних Web-сервісів - забезпечити комунікацію між додатками, Семантичний Web вирішує більш складну проблему - підвищити цінність інформації, яку можна знайти в Internet. Зробити це можна, побудувавши мережу, яка логічно складається не з розрізнених документів, а з описів реальних об'єктів та їх взаємовідносин з іншими об'єктами.

2.9 Реалізації семантичного WEB

Вже сьогодні практично всі відомі компанії рівня IBM, Adobe або Sun Microsystems, активно використовують технологію Семантичного Web у своїх продуктах для вирішення завдань управління даними.

Компанія Microsoft інвестує сотні мільйонів доларів у проект взаємодіючих мережевих ресурсів .NET, який відображає їхнє уявлення про найближче майбутнє Internet. Створювана система дозволяє проводити автоматизований обмін мережевими ресурсами між окремими програмами, додатками, базами даних, користувачами, ґрунтуючись на XML, як на ключовий технології.

У Європі ведеться проект, подібний семантичного Web, - "Мережа знань", Knowledge Web (<http://kw.dia.fi.upm.es/semanticportal/jsp/frames.jsp>). Ця мережа орієнтована на потреби інформаційних технологій в промисловості, науці та освіті, а Семантичний Web (підтримуваний і в Європі - SWAD-Europe, www.w3.org/2001/sw/Europe/) більше розрахований на електронну комерцію і спрощення роботи користувачів мережі Інтернет.

Нещодавно в рамках ідеології Семантичного Web була розроблена в School of Electronics & Computer Science (ECS) Університету Саутгемптона була розроблена система mSpace. Програмне забезпечення цієї системи являє собою набір потужних інструментів, що дозволяють збирати дані з різних джерел і організувати інформацію по категоріях і дають можливість користувачеві вільно орієнтуватися в ній.

Розробники наводять наступний приклад. Наприклад, якщо в Google набрати "класична музика", то пошуковець видасть посилання на сайти, так чи інакше стосуються класичної музики. Якщо ж шукати "класичну музику" на mSpace, то буде виданий список композицій, які можна тут же завантажити. Інший приклад - за запитом "Гаррі Поттер" користувач отримає не просто набір посилань, а відсортований звіт, в якому частина посилань буде лежати в графі "фільми", інша частина - в колонці "книги", а третя - в колонці "рецензії".

Семантичний Web надасть користувачеві можливість вибирати, в якому напрямку досліджувати інформацію, а не просто видавати саме релевантне за загальним алгоритмом.

2.10 Використання семантичного WEB в наукових дослідженнях

Бази знань наукових досліджень будуються на основі різних зв'язків. Підтримка семантичних зв'язків явним чином призводить до утворення багатопшарової семантичної структури контенту електронної бібліотеки, кожен шар якої відповідає деякій властивості семантичних зв'язків. Така структура може формуватися на основі онтології зв'язків і може служити джерелом інформації для проведення якісно нових наукометричних вимірювань, для дослідження структурних властивостей корпусу знань у різних галузях науки.

В останні роки на основі бібліографічних посилань у публікаціях, що випускаються в авторитетних періодичних виданнях, почали створюватися індекси цитування, які формують бібліометричну статистику. Зв'язки цитування в текстових публікаціях зазвичай представляються неструктурованим чином у вигляді списку використаної літератури.

Поряд зі зв'язками цитування між документами наукових електронних бібліотек існують різноманітні інші семантичні зв'язки. Наприклад, зв'язок може вказувати, що її цільовий документ містить наукові результати, що базуються на результатах, описаних у вихідному документі зв'язку, або що у вихідному документі зв'язку спростовується результат, викладений у її цільовому документі. Зв'язок може також вказувати, що вихідний її документ є новою редакцією цільового документа або що він являє собою його складову частину, наприклад, анотацію. Існує велика різноманітність семантичних зв'язків, які можна при необхідності підтримувати між документами бібліотеки.

База знань термінів, законів, визначень, понять, зв'язків з іншими науками дозволяють, наприклад, створювати такі цикли:

- знання певної наукової діяльності зібрані і об'єднані в базу знань
- дані перевіряються на цілісність, запити до них та автоматизовані міркування призводять до гіпотез
- гіпотези можуть допомогти в розробці нових експериментів
- експериментування генерує нові дані і, можливо, підтверджує або заперечує гіпотезу
- дані, отримані після цього будуть інтегровані в базу знань, створюючи циклічний процес

2.11 Висновки

Отже, розвиток технологій семантичних Веб-сервісів дасть можливість створювати єдине уніфіковане представлення семантичного Веб-сервісу в різних застосуваннях, дозволить точно знаходити необхідну інформацію, спрощувати корпоративну інтеграцію та інтеграцію мережевих додатків в розподіленому Веб-просторі. Семантика забезпечує автоматизацію динамічного пошуку, композиції і виконання Веб-сервісів. Акценти зміщуються від синтаксичної до семантичної інтероперабельності на основі підходів прийняття рішень. Наприклад, для наукових статей підтримка семантичних зв'язків явним чином призводить до утворення багат шарової семантичної структури контенту електронної бібліотеки, кожен шар якої відповідає деякій властивості семантичних зв'язків. Така структура може формуватися на основі онтології зв'язків і може служити джерелом інформації для проведення якісно нових наукометричних вимірювань, для дослідження структурних властивостей корпусу знань у різних галузях науки. Виведення гіпотез, їх підтвердження та генерування нових даних, що в свою чергу знову записуються в базу знань, може дати науці новий шлях для отримання нових відкриттів.

Основною мотивацією розвитку семантичних методологій є:

- Постійно зростаючі вимоги ринку ІТ-проектів до високої швидкості розробки рішень, гнучкості, простоті в обслуговуванні, що визначає необхідність нових, заснованих на стандартах архітектур і методологій;
- Вимоги ІТ-проектів до високого ступеня інтеграції даних і додатків, управління бізнес-процесами корпорацій, ефективність використання ресурсів;
- Зростаючі витрати на перевірку та тестування додатків вимагають збільшення продуктивності розробки з використанням засобів автоматизації.

Для підтримки інтеграції розробляються мови моделювання, які еволюціонують від синтаксичної орієнтації та значної частки людської участі до семантичної і машинно-інтерпретуємої орієнтації. З появою мов семантичного моделювання, таких як Resource Description Framework (RDF) і Web Ontology Language (OWL), з'явилася необхідна основа для інтеграції додатків на основі семантики. Семантичні Веб-сервіси, що використовують мови моделювання OWL і WSMO, спрямовані на реалізацію можливості автоматизації процесів розробки та інтеграції додатків.[4]

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СЕМАНТИЧНОГО ВЕБ-ДОДАТКУ

3.1 Вибір технологій

Для програмної реалізації Семантичного Web-додатку було обрано мову програмування PHP. Це скриптова мова програмування, що була створена для генерації HTML-сторінок на стороні веб-сервера. Для роботи з даною мовою був піднятий локальний сервер. Використовується WAMP - набір (комплекс) серверного програмного забезпечення (Apache, MySQL, PHP).

Для реалізації інтерфейсу використовуються такі технології як HTML, CSS, Javascript, JQuery. На стороні серверу додаток має архітектуру MVC.

Були створені основні класи, контролери і відображення. Модель-вигляд-контролер (або Модель-вид-контролер, англ. Model-view-controller, MVC) — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування. Застосовується для відокремлення даних (модель) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача. Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності. [5]

Архітектурний шаблон Модель-Вид-Контролер (MVC) поділяє програму на три частини. У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за представлення цих даних користувачеві. Контролер

(Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

MVC поділяє цю частину системи на три самостійні частини: введення даних, компонент обробки даних і виведення інформації. Модель, як вже було відмічено, інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.

Для роботи з RDF-файлами було обрано бібліотеку Graphite PHP Linked Data Library. Вона безкоштова, відкритий код, побудована на основі ARC2.

Для написання SPARQL запитів використовується бібліотека sparqllib.php.

3.2 Архітектура додатку.

Як було зазначено вище, додаток розроблений як MVC. Структура директорій:

```
/cache  
/controllers  
    /main  
/css  
/img  
/js  
/libs  
    /arc  
    /Graphite  
/model  
/view
```

Усі запити на сервер обробляє файл `index.php`, у якому створюється екземпляр сінглтон класу, який в свою чергу підтягує всі необхідні файли, обробляє строку запиту, викликає необхідний контролер, що вибирає дані та віддає клієнту сформовану відповідь у вигляді HTML сторінки. Архітектура зображена на рисунку 3.1.

Використовуючи бібліотеку Graphite PHP в додатку можливо шукати діставати інформацію з RDF-файлів. Також було під'єднано бібліотеку `sparqllib.php`, за допомогою якої можливо з'єднуватися з точкою доступу SPARQL, та писати запити до неї, знаходити необхідну інформацію. Для використання цих двох бібліотек написані моделі. Загалом в них написані готові запити для демонстрації роботи додатку.

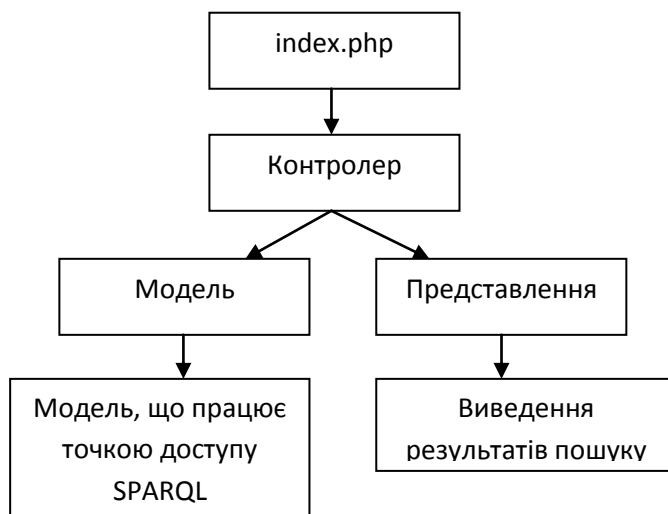


Рисунок 3.1 – Архітектура проекту.

3.3 Інформація про використані бібліотеки

Для роботи з RDF-файлами використовується бібліотека Graphite PHP. Це безкоштовна, з відкритим кодом бібліотека, що була розроблена для того, щоб зробити роботу з RDF простіше і зрозуміліше. Побудована на останній версії ARC2, щоб легко і дуже швидко працювати з даними RDF, без зайвих клопіт с базами даних. Вона не призначена, щоб бути масштабованою.

На сайті розробника детально описано основний функціонал, приклади та документація методів.

Ось мінімальний приклад, з якого можна почати. Отримання назви університету (Рисунок 3.1).

```
public function index(){
    $this->setTitle("Examples");

    $graph = new Graphite();
    $graph->load( "http://id.southampton.ac.uk/" );
    $result = $graph->resource( "http://id.southampton.ac.uk/" )->get( "foaf:name" );
    $this->loadView('rdfExample', array('result' => $result));
}
```

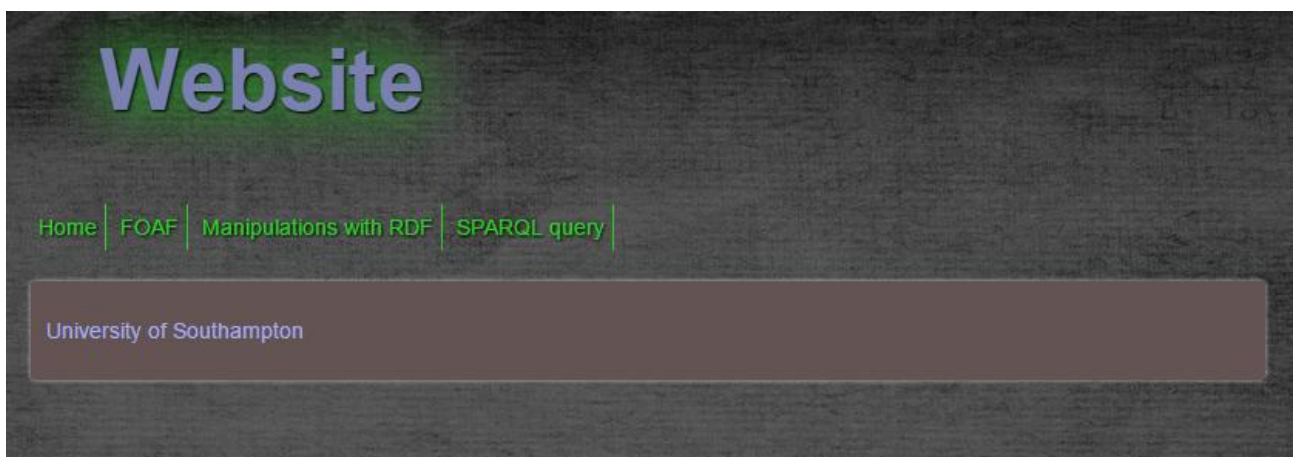


Рисунок 3.2 – Отримання назви університету з відповідного RDF-файлу.

За замовчуванням, графіт визначає купу часто використовуваних неймспейсів: foaf, dc, dct, rdf, rdfs, owl, xsd, cc, bibo, skos, geo, sioc. Але можна додати свої власні, щоб зробити код більш читабельним.

Щоб побачити всі вхідні і вихідні зв'язки з URI, використовується метод відладки. Ліві стрілки вказують на вхідні зв'язки. Можна вивести об'єкт Graphite, щоб побачити все (Рисунок 3.3). Метод all() повертає масив відповідних значень. Використання "-" призводить до зворотних відносин, наприклад якщо написати так "-foo:Child", то це відповідатиме речі, що є дитиною, а не дитиною від нього (Рисунок 3.4).

```
public function index(){
    $this->setTitle("Examples");

    $graph = new Graphite();
    $uri = "http://data.ordnancesurvey.co.uk/id/postcodeunit/SO171BJ";
    $graph->load( $uri );
    $result = $graph->resource( $uri )->dump();

    $this->loadView('rdfExample', array('result' => $result));
}
```

Home | FOAF | Manipulations with RDF | SPARQL query

http://data.ordnancesurvey.co.uk/id/postcodeunit/SO171BJ

- rdf:type → http://data.ordnancesurvey.co.uk/ontology/postcode/PostcodeUnit
- http://data.ordnancesurvey.co.uk/ontology/spatialrelations/within → http://data.ordnancesurvey.co.uk/id/postcodearea/SO, http://data.ordnancesurvey.co.uk/id/postcodesector/SO171, http://data.ordnancesurvey.co.uk/id/postcodedistrict/SO17
- http://data.ordnancesurvey.co.uk/ontology/postcode/LH → "E18000009"
- http://data.ordnancesurvey.co.uk/ontology/postcode/nhsRegionalHA → http://statistics.data.gov.uk/id/statistical-geography/E19000002
- http://data.ordnancesurvey.co.uk/ontology/postcode/positionalQualityIndicator → http://data.ordnancesurvey.co.uk/ontology/postcode/pqi/10
- geo:long → "-1.395675"^^xsd:decimal
- http://data.ordnancesurvey.co.uk/ontology/spatialrelations/northing → "115145.00"^^xsd:decimal
- http://data.ordnancesurvey.co.uk/ontology/spatialrelations/easting → "442562.00"^^xsd:decimal
- geo:lat → "50.934180"^^xsd:decimal
- skos:notation → "SO17 1BJ"^^http://data.ordnancesurvey.co.uk/ontology/postcode/Postcode
- http://data.ordnancesurvey.co.uk/ontology/postcode/RH → "E19000002"
- http://data.ordnancesurvey.co.uk/ontology/postcode/ward → http://data.ordnancesurvey.co.uk/id/7000000000017711
- http://data.ordnancesurvey.co.uk/ontology/postcode/country → http://data.ordnancesurvey.co.uk/id/country/england
- http://data.ordnancesurvey.co.uk/ontology/postcode/nhsHA → http://statistics.data.gov.uk/id/statistical-geography/E18000009
- http://data.ordnancesurvey.co.uk/ontology/postcode/PQ → "10"
- http://data.ordnancesurvey.co.uk/ontology/postcode/district → http://data.ordnancesurvey.co.uk/id/7000000000037256
- rdfs:label → "SO17 1BJ"
- ← is foaf:primaryTopic of ← http://data.ordnancesurvey.co.uk/doc/postcodeunit/SO171BJ

Рисунок 3.3 – Об'єкт Graphite.


```

public function index(){
    $this->setTitle("Examples");

    $graph = new Graphite();
    $graph->ns( "org", "http://www.w3.org/ns/org#" );
    $uri = "http://id.southampton.ac.uk/org/F2";
    $graph->load( $uri );
    $result = $graph->resource( $uri )
        ->all( "org:hasSubOrganization" )
        ->sort( "rdfs:label" )
        ->getString( "rdfs:label" )
        ->join( ", " ).".\n";

    $this->loadView('rdfExample', array('result' => $result));
}

```

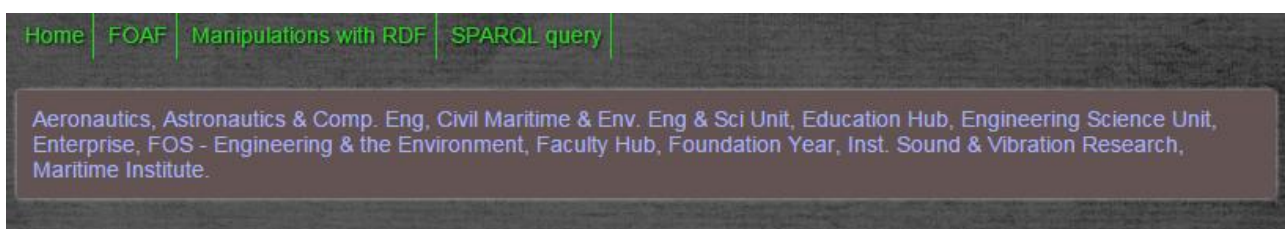


Рисунок 3.4 – Отримання списку ярликів відсортованих по алфавіту.

Перехід за посиланням на інше джерело даних зробити легко завдяки бібліотеці ARC2. Тим не менш, можна встановити ARC2 так щоб використовувалися не абсолютні посилання (Рисунок 3.5).

Розглянемо бібліотеку для викликів SPARQL запитів з PHP - sparqllib.php. Вона ігнорує інформацію про мову і типи даних, щоб зробити себе максимально схожою з звичайними бібліотеками PHP SQL.

Бібліотека включає в себе функції, що дуже схожі на функції mysql_ для зручності (Рисунок 3.6). Також можна використовувати об'єктно-орієнтований стиль.

На основі цих двох бібліотек автор створив декілька веб-додатків. Можна виділити два з них – це RDF Browser та SPARQL Browser. За допомогою RDF браузера можна знаходити і продивлюватись зміст RDF файлів по їх URI (Рисунок 3.7, Рисунок 3.8). SPARQL браузер дозволяє чи по заданій точці

доступу показати усі типи файлів(rdf:types) що має, чи показати усі триплети що має URI, чи показати усі триплети, в яких є строчка тексту введена в формі пошуку (Рисунок 3.9, Рисунок 3.10).

```

public function index(){
    $this->setTitle("Examples");

    $person_uri = "http://eprints.ecs.soton.ac.uk/id/person/ext-1248";

    $graph = new Graphite();
    $graph->cacheDir( "C:\wamp\www\semantic\cache" );
    $graph->load( $person_uri );

    $person = $graph->resource( $person_uri );
    $result = "<h3>".$person->link()."</h3>";

    foreach( $person->all( "owl:sameAs" ) as $sameas ){
        $result .= "<div>sameAs: ".$sameas->link()."</div>";
    }

    $result .= $this->showPersonInfo("Before",$person);
    # Проходить по посиланням і підгружає в граф
    $person->loadSameAs();
    $result .= $this->showPersonInfo("After",$person);

    $this->loadView('rdfExample', array('result' => $result));
}

private function showPersonInfo($title,$person)
{
    $html = "<h4>$title</h4>";
    $html .= "<div><b>name:</b> ".$person->all( "foaf:name" )->join( ", ")."</div>";
    $html .= "<div><b>phone:</b> ".$person->all( "foaf:phone" )->prettyLink()->join( ", ")."</div>";
    $html .= "<div><b>homepage:</b> ".$person->all( "foaf:homepage" )->link()->join( ", ")."</div>";

    return $html;
}

```



Рисунок 3.5 – Приклад отримання даних по посиланням.

```

public function sparql(){
    $db = sparql_connect( "http://rdf.ecs.soton.ac.uk/sparql/" );
    if( !$db ) { print sparql_errno() . ": " . sparql_error(). "\n"; exit; }
    sparql_ns( "foaf", "http://xmlns.com/foaf/0.1/" );

    $sparql = "SELECT * WHERE { ?person a foaf:Person . ?person foaf:name ?name } LIMIT 5";
    $result = sparql_query( $sparql );
    if( !$result ) { print sparql_errno() . ": " . sparql_error(). "\n"; exit; }

    $fields = sparql_field_array( $result );

    $resultHtml = "<p>Number of rows: ".sparql_num_rows( $result )." results.</p>";
    foreach( $fields as $field )
    {
        $resultHtml .= $field . " ";
    }
    $resultHtml .= "<ul>";
    while( $row = sparql_fetch_array( $result ) )
    {
        $resultHtml .= "<li>";
        foreach( $fields as $field )
        {
            $resultHtml .= $row[$field] . " ";
        }
        $resultHtml .= "</li>";
    }
    $resultHtml .= "</ul>";

    $this->loadView('rdfExample', array('result' => $resultHtml));
}

```

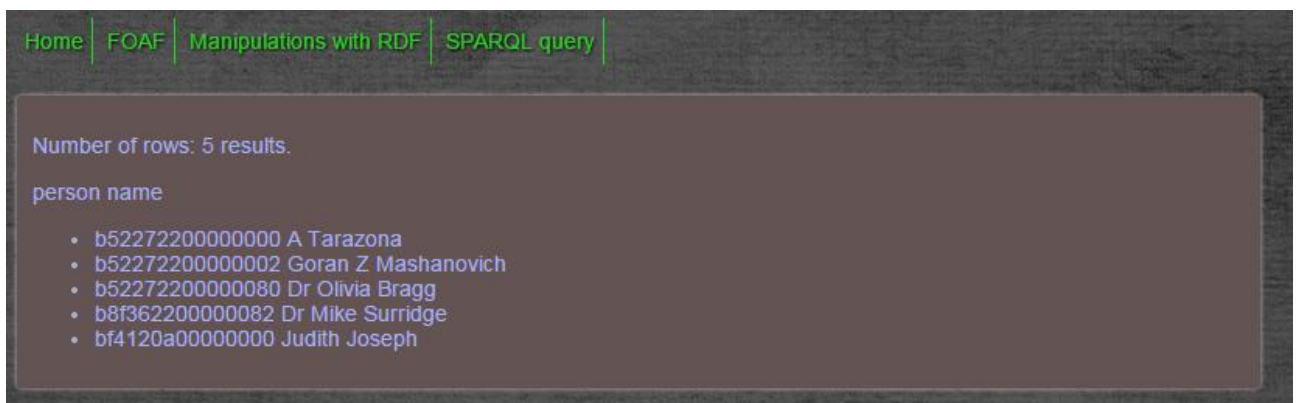


Рисунок 3.6 – Приклад отримання даних по посиланням.

RDF Browser | [SPARQL Browser](#)

Quick and Dirty RDF browser

URI:

Рисунок 3.7 – RDF браузер.

<http://data.totl.net/playingcards/> [\[view\]](#)

Q&D RDF Browser is powered by [Graphite](#) and [ARC2](#) and hosted by [ECS](#) at the [University of Southampton](#).

689 Triples

Ace of Clubs http://data.totl.net/playingcards/card/AceOfClubs → rdf:type → http://data.totl.net/cards/Card , http://data.totl.net/cards/SuitedCard , http://data.totl.net/cards/ValuedCard → foaf:name → "Ace of Clubs" → http://data.totl.net/cards/suit → Clubs → http://data.totl.net/cards/numericValue → "1", "14" → http://data.totl.net/cards/value → Ace → http://data.totl.net/cards/colour → Black → http://data.totl.net/cards/symbol → "A♣" ← is http://data.totl.net/cards/hasCard of ← Classic Playing Cards	a Card
Ace of Diamonds http://data.totl.net/playingcards/card/AceOfDiamonds → rdf:type → http://data.totl.net/cards/Card , http://data.totl.net/cards/SuitedCard , http://data.totl.net/cards/ValuedCard → foaf:name → "Ace of Diamonds" → http://data.totl.net/cards/suit → Diamonds → http://data.totl.net/cards/numericValue → "1", "14" → http://data.totl.net/cards/value → Ace → http://data.totl.net/cards/colour → Red → http://data.totl.net/cards/symbol → "A♦" ← is http://data.totl.net/cards/hasCard of ← Classic Playing Cards	a Card
Ace of Hearts http://data.totl.net/playingcards/card/AceOfHearts	a Card

Рисунок 3.8 – Результаты RDF браузера по введённому URI.

[RDF Browser](#) | [SPARQL Browser](#)

Quick and Dirty SPARQL browser

Endpoint:

URI:

Text search:

Рисунок 3.9 – SPARQL браузер.

<http://rdf.ecs.soton.ac.uk/sparql/>

Q&D SPARQL Browser is powered by [Graphite](#) and [ARC2](#) and hosted by [ECS](#) at the [University of Southampton](#).

22 results.

http://eprints.ecs.soton.ac.uk/id/document/4316	Color Edge Detection Hardware based on Geometric Algebra. (PDF)	
http://eprints.ecs.soton.ac.uk/id/document/44736	Partial Discharge Simulation for a High Voltage Transformer Winding using a Model based on Geometrical Dimensions (Image (PNG))	
http://eprints.ecs.soton.ac.uk/id/document/9219	Towards the evolutionary emergence of increasingly complex advantageous behaviours (Image (PNG))	
http://eprints.ecs.soton.ac.uk/id/eprint/334	Parameterisation of Simple Geometrical Lambertian Surfaces using Photometric Stereo	Article , EPrint , ConferenceItem , EPrint , AcademicArticle
http://eprints.ecs.soton.ac.uk/id/eprint/9010	The Numerical Simulation of VLF Chorus and Discrete Emissions Observed on the Geotail Satellite using a Vlasov Code	Article , EPrint , ArticleEPrint , AcademicArticle
http://eprints.ecs.soton.ac.uk/id/person/ext-166abd877e5e412e9a19e8312d481127	George Chin Jr.	
http://eprints.ecs.soton.ac.uk/id/person/ext-23806	George Cushen	
http://eprints.ecs.soton.ac.uk/id/person/ext-5400c5890df577425a2a6ff4c1601a18	George Agapiou	

Рисунок 3.10 – Результаты SPARQL браузера по введённым данным у форму.

3.4 Функціонал створеного додатку

В даному додатку було реалізовано приклади використання точок доступу SPARQL для отримання необхідної інформації с онтологій та RDF-файлів за допомогою запитів SPARQL. Програма була створена з метою ознайомлення з принципами, методиками і технологіями розробки семантичних веб-додатків.

З головної сторінки (Рисунок 3.11) ми можемо перейти у вкладку “FOAF” (Рисунок 3.12), де можна обрати дві опції:

- Persons
- Find person with name like your quey string

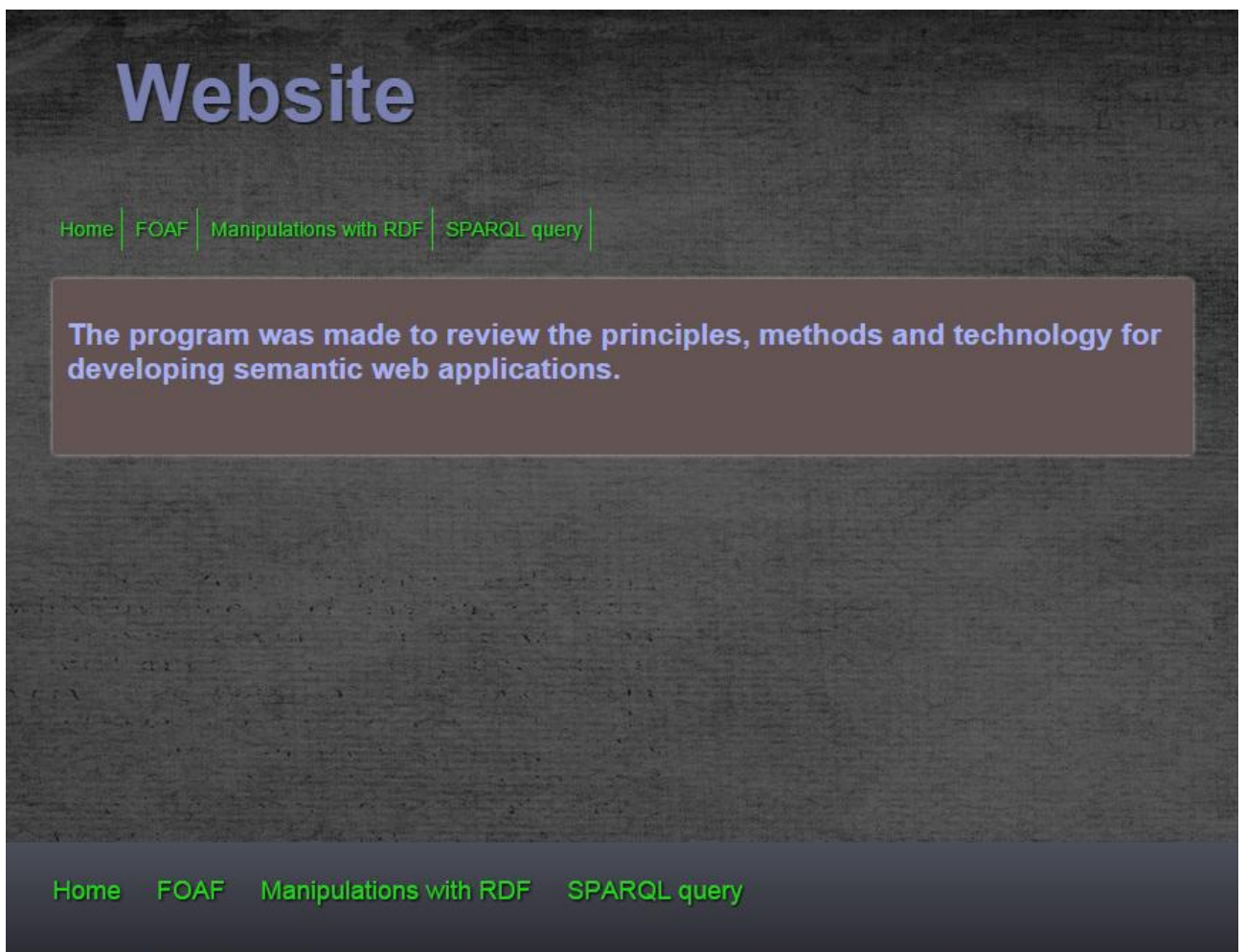


Рисунок 3.11 – Головна сторінка додатку.



Рисунок 3.12 – Сторінка FOAF.

Перша опція реалізовує SPARQL запит:

```
"SELECT * WHERE { ?person a foaf:Person . ?person foaf:name ?name } LIMIT 100"
```

Результатом є список до 100 персон: таблиця з ідентифікатором особи та іменем (Рисунок 3.13).

The screenshot shows the same website as in Figure 3.12, but the main content area displays the results of a SPARQL query. The results are presented as a table with two columns: "Person" and "Name".

Person	Name
b52272200000000	A Tarazona
b52272200000002	Goran Z Mashanovich
b52272200000080	Dr Olivia Bragg
b8f362200000082	Dr Mike Surridge
bf4120a00000000	Judith Joseph
bf4120a00000002	pz
bf4120a00000082	ama04r
bf6120a00000080	Lizhe Tan

Рисунок 3.13 – Результат запити.

Друга опція дозволяє нам знайти персон з іменем, в яке входить текст запиту (Рисунок 3.14). Наприклад, якщо ввести ‘and’, то в результаті отримаємо усіх тих, у коо в імені зустрічається ‘and’(Рисунок 3.15).

Рисунок 3.14 – Сторінка FOAF.

Person	Name
b152272200000086	Prof Sandor M. Veres
b1fb120a00000006	Paul Andrews
b5fa120a00000017	Srinandan Dasmahapatra
b7f9120a0000001f	Prof. Andy Adamatzky (Bristol)
b7fb120a0000001e	Prof. Maria.Evandrou
b9f8120a00000026	Professor Philip Newland (Neuroscience)
bafa120a00000029	Srinandan Dasmahapatra
bdf5120a000000b4	Prof Andrew Brown
b1efa120a000000fa	Andrew Dainty
b1f5227220000007f	Dr Andrew Black

Рисунок 3.15 – Результат пошуку.

Обидві опції використовували точку доступу <http://rdf.ecs.soton.ac.uk/sparql/> та ідентифікатор ресурсу <http://xmlns.com/foaf/0.1/>.

Також, якщо перейти у вкладку ‘SPARQL query Artists’, ми отримаємо результат наступного SPARQL запиту, в якому ми шукаємо діячів культури та мистецтва(включаючи акторів, представників бойових мистецтв), засновників компаній, що родилися після 1939 року (Рисунок 3.16).[6]

Artists list:

Company url	Corporation	Founder URL	Founder Birth
http://dbpedia.org/resource/Basement_Comics	Basement Comics	http://dbpedia.org/resource/Budd_Root	1958-01-01+02:00
http://dbpedia.org/resource/Mam_Tor_Publishing	Mam Tor Publishing	http://dbpedia.org/resource/Liam_Sharp	1968-05-02+02:00
http://dbpedia.org/resource/Hand_Made_Films	HandMade Films	http://dbpedia.org/resource/George_Harrison	1943-02-25+02:00
http://dbpedia.org/resource/Eio_Books	Eio Books	http://dbpedia.org/resource/Ki_Loingfellow	1944-12-09+02:00
http://dbpedia.org/resource/Pop!_Goes_the_Icon	Pop! Goes the Icon	http://dbpedia.org/resource/Pj_Peretz	1976-08-04+02:00
http://dbpedia.org/resource/Pottermore	Pottermore	http://dbpedia.org/resource/J._K._Rowling	1965-07-31+02:00
http://dbpedia.org/resource/Artists_Music_Guild	Artists Music Guild, Inc.	http://dbpedia.org/resource/David_L_Cook	1968-11-11+02:00
http://dbpedia.org/resource/Broadway_Comics	Broadway Comics	http://dbpedia.org/resource/Jim_Shooter	1951-09-27+02:00
http://dbpedia.org/resource/Defiant_Comics	Defiant Comics	http://dbpedia.org/resource/Jim_Shooter	1951-09-27+02:00
http://dbpedia.org/resource/HGGC	HGGC	http://dbpedia.org/resource/Greg_Benson	1968-01-23+02:00

Рисунок 3.16 – Сторінка SPARQL query Artists.

Дана інформація формується кодом:

```
public function artists(){
    $this->setTitle("Artists List");
    $this->addCss('/css/users.css');

    Index::sparqlDbConnect("http://dbpedia.org/sparql");
    Index::$DB->ns("rdf","http://www.w3.org/1999/02/22-rdf-syntax-ns#");
    Index::$DB->ns("dbpedia-owl","http://dbpedia.org/ontology/");
    Index::$DB->ns("xsd","http://www.w3.org/2001/XMLSchema#");
```



```

$result = Index::$DB->query('SELECT * WHERE
{
?companyURL rdf:type dbpedia-owl:Company      ;
dbpprop:companyName ?corporation              ;
dbpedia-owl:foundedBy ?founderURL             .
?founderURL dbpedia-owl:birthDate ?founderBirth .
FILTER (?founderBirth > "1940-03-10"^^xsd:date ) .
{ ?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
');

if( !$result ) { print Index::$DB->errno() . ": " . Index::$DB->error(). "\n";
exit; }
$fields = $result->field_array( $result );

$persons = array();
while( $row = $result->fetch_array() )
{
    foreach( $fields as $field )
    {
        $person[$field] = $row[$field];
    }

    persons[] = $person;
}

$this->loadView('artists', array('persons' => $persons));
}

```

У даному прикладі[6] використовуються <http://dbpedia.org/sparql> як точка доступу. Неймспейси:

- <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- <http://dbpedia.org/ontology/>
- <http://www.w3.org/2001/XMLSchema#>

DBpedia - це проект створення структурованої копії Вікіпедії, що постійно оновлюється, представленої в мові RDF. Для того, щоб створити DBpedia, розробники написали безліч програм-хробаків, які день і ніч витягають з вікіпедії всю структуровану інформацію, яку вона може запропонувати: Категорії, Посилання 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).[6]

Наприклад, сторінка Леонардо Да Вінчі може бути представлена наступним чином (Рисунок 3.17):

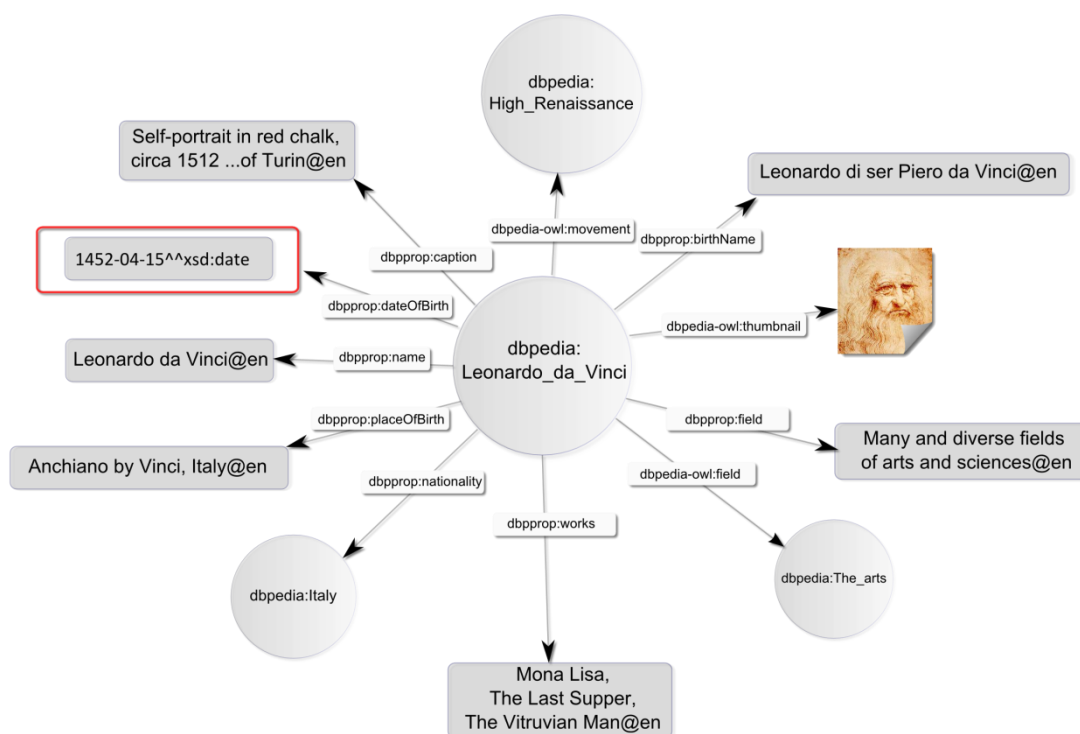


Рисунок 3.17 – Граф, отриманий з Вікіпедії. Кружками позначені ідентифікатори ресурсів, прямокутниками - літерали.[6]

Отже, DBpedia велика, дані в ній містять безліч як внутрішніх посилань, так і посилань на зовнішні сховища. Витяг даних відбувається автоматично, тому не варто розраховувати на те, що отримані RDF-трійки будуть без шкідливих домішок і помилок парсинга.

Точка доступу SPARQL - це служба, що підтримує протокол запитів SPARQL. Точка доступу дозволяє користувачеві робити запити до бази знань. Сервер обробляє запит і повертає відповідь в деякому, зазвичай машинно-розуміючому, форматі. Таким чином, точки доступу SPARQL в першу чергу є

API до баз знань, а представлення результатів має бути реалізовано програмним забезпеченням запитуючої сторони.[6]

Розрізняють два види точок доступу: загального призначення та локальні.

Точки доступу загального призначення можуть виробляти запити з будь-яким зазначеним RDF-документами, що знаходяться в Мережі. А локальні точки доступу здатні отримувати дані тільки від одного ресурсу.[6]

3.5 Висновки

Отже, під час пошуку інформації та розуміння концепції семантичних веб-додатків було створено власну програму з метою ознайомлення з принципами, методиками і технологіями розробки таких веб-додатків. Було реалізовано приклади використання декількох точок доступу SPARQL для отримання інформації с онтологій та RDF-файлів в залежності від запитів SPARQL. Були використані бібліотеки для роботи з SPARQL запитами і RDF-файлами. Практична реалізація семантичного додатку забезпечило краще розуміння необхідності використання такої концепції побудови інформації. Такий підхід організації даних та доступу до них дає більше можливостей ніж поширена зараз конструкція СУБД + інтерфейс. Але є також і недоліки та труднощі в реалізації семантичного вебу.

4 ОХОРОНА ПРАЦІ

Програміст – людина, що займається програмуванням, виконує розробку програмного забезпечення (в простіших випадках – окремих програм) для програмованих пристроїв. Прикладами таких пристроїв є: настільні персональні комп'ютери, мобільні телефони, сервери, суперкомп'ютери, мікроконтролери та промислові комп'ютери. Робота розробника програмного забезпечення характеризується значною розумовою напругою і нервово-емоційним навантаженням, високою напруженістю зорової роботи і достатньо великим навантаженням на кисті рук при роботі з клавіатурою. Тому великого значення набувають раціональна конструкція і розташування елементів робочого місця, наявність достатнього робочого простору, нормального освітлення, допустимого рівня шуму. Також актуальними є питання щодо енергобезпеки та пожежної безпеки на робочому місці.

4.1 Характеристика робочого місця

Проаналізуємо робоче приміщення, зображене на рис.1. Приміщення має одnobічне природне освітлення та загальне штучне освітлення. Вікно обладнане завісками, орієнтовано на північ, площа засклення 40%. Стіни обклеєні світлими шпалерами, стеля біла, підлога вкрита світлим паркетом з природнього матеріалу.

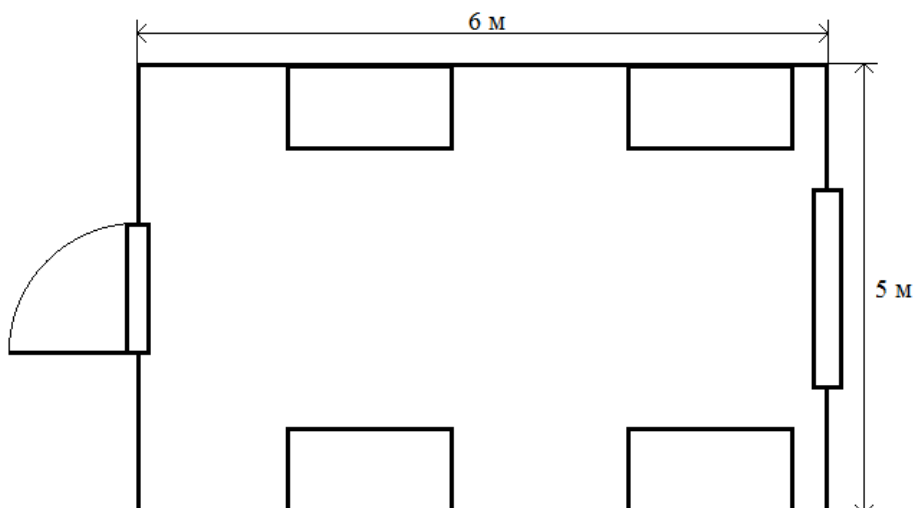


Рис. 1. План робочого приміщення

Загальна площа приміщення складає 30 м², його ширина становить 5 метрів, довжина – 6 метрів, висота стелі – 3 метри. В приміщенні працює чотири людини.

Виходячи з того, що в приміщенні працює чотири людини, можна обрахувати фактичні та нормативні значення площі та об'єму приміщення з розрахунку на одне робоче місце. Отримані результати наведені в табл. 1.

Таблиця 1

Фактичні та нормативні значення площі та об'єму приміщення з розрахунку на одне робоче місце

Параметр приміщення	Нормативний	Фактичний
Площа, м ²	6 і більше	7.5
Об'єм, м ³	20 і більше	22.5

Виходячи з даних, наведених в табл. 1, можна зробити висновок, що розміри приміщення задовольняють існуючим вимогам [7].

4.2 Аналіз умов праці

4.2.1 Напруженість праці програміста

Виходячи з характеру розробленого програмного забезпечення та згідно з [8] робота користувача ЕОМ за показниками напруженості трудового процесу відноситься до наступних категорій:

- за показником інтелектуального навантаження – напружена 1 ступеня (класи 1-4, в середньому клас 3: частково творча і частково здійснюється за алгоритмами, вимагає оцінок взаємопов'язаних параметрів);
- за емоційним навантаженням – оптимальна (клас 1: відповідальність за окремі елементи роботи і відсутність ризиків для життя і безпеки);

- за монотонністю навантажень – напружена 2 ступеня (клас 4: робота на ЕОМ характеризується постійним активним виконанням різних операцій);
- за сенсорним навантаженням – напружена 2 ступеня (клас 4: робота на ЕОМ більше 4 годин на добу);
- за режимом праці – допустима (клас 2: близько 8 годин на добу).

Отже, характер робіт в середньому відповідає класу 3 (напружена робота 1 ступеня), тому необхідно періодично та в достатній кількості вживати заходів щодо збереження здоров'я: розробити та виконувати комплекс вправ, забезпечити здорове харчування під час періодів відпочинку. Для розробників програмного забезпечення рекомендовано проводити перерви для відпочинку тривалістю 15 хвилин через кожну годину роботи.

4.2.2 Мікроклімат

Для постійних робочих місць, якими є робочі місця програмістів, встановлені оптимальні параметри мікроклімату (згідно з [9]), а при неможливості їх дотримання використовують допустимі параметри. Небезпечним фактором є мікроклімат приміщення, хоча приміщення знаходиться не з сонячної сторони, влітку у приміщенні все одно дуже жарко, а взимку холодно.

Через те, що приміщення належить до І(а) категорії (виконуються легкі фізичні роботи), повинні дотримуватися вимоги наведені у табл. 2.

Таблиця 2

Параметри мікроклімату для приміщень з ПК

Період року	Параметр мікроклімату	Норма	Реальна величина

Холодний	Температура повітря в приміщенні	22...24 °С	24 °С
	Відносна вологість	40... 60%	55%
	Швидкість руху повітря	0,1 м/с	0,1 м/с
Теплий	Температура повітря в приміщенні	23...25 °С	25 °С
	Відносна вологість	40...60%	55%
	Швидкість руху повітря	0,1 м/с	0,1 м/с

Для створення й автоматичної підтримки в приміщенні оптимальних значень температури, вологості, чистоти і швидкості руху повітря незалежно від зовнішніх умов, у холодний час року використовується водяне опалення, у теплий час року застосовується кондиціонування повітря. Для охолодження приміщення використовується кондиціонер *ORION MCH-09* з потужністю 2,75 кВт та рівнем шуму 39 дБА. Вибір кондиціонеру проводився по потужності з урахуванням необхідної продуктивності по об'єму повітря, що подається до кімнати. Під час розрахунку потужності охолодження враховуються всі можливі теплотоки та тепловиділення у приміщенні при найбільш несприятливих умовах, основними параметрами при розрахунку мають бути розмір вікна та кількість комп'ютерів у приміщенні.

Таблиця 3

Норми подачі свіжого повітря в приміщення з ПК

Характеристика об'єму	Об'ємна витрата свіжого повітря, що подається в приміщення, на одну людину в годину
Менше 20 м ³ на людину	Не менше 30
20 – 40 м ³ на людину	Не менше 20
Більше 40 м ³ на людину	Може використовуватися природна вентиляція

4.2.3 Шум

Згідно з [10] допустимий рівень шуму складає 50 дБА. Основними

джерелами шуму в приміщенні є кондиціонер *ORION MCH-09* з максимальним рівнем шуму 39 дБА, а також чотири ЕОМ з системами охолодження, максимальний рівень шуму кожної з яких становить 20 дБА.

Сумарний шум L від п'яти джерел обрахуємо за формулою:

$$L = 10 \lg \left(10^{0,1 \cdot 39} + 10^{0,1 \cdot 20} + 10^{0,1 \cdot 20} + 10^{0,1 \cdot 20} + 10^{0,1 \cdot 20} \right).$$

Отже, фактичний сумарний рівень шуму складає 39,21 дБА, що (згідно з [10]) задовольняє вимогам (до 50 дБА для програмістів ЕОМ). Тому додаткові заходи для зниження рівня шуму не потребуються.

4.2.4 Освітлення робочого місця

Робота, що виконується з використанням обчислювальної техніки потребує належного освітлення. Природне освітлення є найбільш прийнятним для роботи, але його не завжди достатньо.

Відповідно до [11] в приміщеннях для роботи з ЕОМ повинне використовуватися як природне так і штучне освітлення. Природне освітлення забезпечує вікно загальна площа якого складає 3,6 м². Воно являється боковим та одностороннім.

Знайдемо коефіцієнт природної освітленості за формулою:

$$i = \frac{S_{Bik}}{S_n} = \frac{3,6}{30} = 0,12,$$

де S_{Bik} – загальна площа вікна, м²; S_n – площа підлоги, м².

Отримане значення 0.12 менше ніж встановлено нормами [11] (коефіцієнт природної освітленості має бути не меншим за 0.15), тобто природного освітлення не вистачає для нормальної роботи. Тому потрібно використовувати ще й штучне освітлення.

В приміщенні використовується загальне штучне освітлення. Воно забезпечується світильниками типу ЛПО з лампами типу ЛБ, потужністю 20 Вт. Для загального освітлення слід застосовувати шість світильників серії ЛПО. Один світильник містить 3 лампи, кожна з яких має світловий потік 1060 лм. Це

задовольнить норму освітленості у 300-500 лк згідно з ДБН В.2.5-28-2006.

4.2.5 Електробезпека

У приміщенні використовується однофазна мережа електропостачання (фазовий, нульовий робочий і нульовий захисний дріт з напругою 220 В і частотою 50 Гц). Проводка проведена прихованим способом, вимикачі і розетки захищені пластмасовими корпусами, світильники розташовані на висоті 3 м.

Розетки типу «Європа» із заземлюючими контактами і мають маркування по напрузі. Заземлюючі контакти розеток мають з'єднання із заземлюючим контуром приміщення. Пристрої, електропроводи і кабелі мають апаратуру захисту від струмів короткого замикання і інших аварійних режимів. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Площа перерізу нульового робочого і нульового захисного провідника більші за площу перерізу фазового дроту. Пристрої підключаються до електромережі тільки за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення, які мають спеціальні контакти для підключення нульового захисного дроту. Обслуговуючий персонал ознайомлений з правилами техніки безпеки при роботі з обчислювальною технікою.

Отже, стан електробезпеки відповідає вимогам [12].

4.2.6 Пожежна безпека

Відповідно до [13] приміщення відноситься до категорії В – пожежонебезпечне, так як на робочому місці розташовані такі пожежонебезпечні матеріали як папір, пластик, пластикові корпуси техніки, меблі.

В нормальних умовах імовірність виникнення пожежі досить низька, а імовірність вибуху – відсутня. Можливими причинами загоряння можуть бути пошкодження та замикання в електромережі та електрообладнанні, а також

порушення правил безпеки при роботі з обладнанням. Для забезпечення пожежного захисту необхідно обладнати приміщення протипожежними засобами.

Відповідно до [14] в приміщенні необхідно встановити один димовий пожежний сповіщувач СПД-3, з розрахунку один на висоту до 3,5 м та загальною площею не більше ніж 86 м². Біля входних дверей повинен розміщуватися план евакуації з приміщення. Робоче приміщення обладнане трьома вуглекислотними вогнегасниками ВВК-3.5, з розрахунку два вогнегасника на 20 м², оскільки загальна площа приміщення складає 30 м².

4.2.7 Випромінювання

Електромагнітне випромінювання може стати причиною як тимчасового погіршення самопочуття людини, зниження продуктивності праці, так і призвести до різних захворювань всіх систем організму людини, серед яких найбільш серйозними є онкологічні захворювання.

Єдиними джерелами шкідливого випромінювання при роботі з ЕОМ є випромінювання ЕПТ-моніторів та природне випромінювання.

Так як монітори ПК вироблені на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновією лампою, то у приміщенні відсутні інфрачервоні, ультрафіолетові та електромагнітні випромінювання.

4.2.8 Ергономіка робочого місця

Висота над рівнем підлоги робочої поверхні, на якій працює користувач, складає 750 мм. Розміри поверхні столу 1400 x 1100 мм². Під столом є простір для ніг з розмірами по глибині 600 мм. Робочий стілець користувача оснащений підйомно-поворотним механізмом (висота регулюється в межах 400 – 550 мм), регулюванням нахилу. Під столом є підставка для ніг.

Екран монітора розташований на відстані 600 мм від очей, клавіатура – на відстані 200 мм від краю стола. Кут нахилу екрана регулюється і в середньому становить 15 градусів по відношенню до користувача.

4.2.9 Інструкція з техніки безпеки

Перед початком роботи на ПК користувач повинен:

- 1) пересвідчитися у цілості корпусів і блоків (обладнання) ПК;
- 2) перевірити наявність заземлення, справність і цілість кабелів живлення, місця їх підключення.

Забороняється вмикати ПК та починати роботу при виявлених несправностях. Під час роботи з ПК необхідно дотримуватися інструкції з експлуатації.

Забороняється:

- 1) замінювати знімні елементи або вузли та проводити ремонт при ввімкненому ПК;
- 2) з'єднувати і роз'єднувати вилки та розетки первинних мереж електроживлення, які знаходяться під напругою;
- 3) знімати кришки, які закривають доступ до струмопровідних частин мережі первинного електроживлення при ввімкненому обладнанні;
- 4) залишати ПК у ввімкненому стані без нагляду.

По закінченні робочого дня:

- 1) відключити електроживлення ПК згідно з інструкцією експлуатації, вийнявши вилку кабелю живлення з розетки;
- 2) впорядкувати робоче місце користувача ПК, прибравши використане обладнання та матеріали у відведені місця;
- 3) про виявлені недоліки у роботі ПК протягом робочого часу необхідно повідомити відповідним особам та спеціалістам.

Залишаючи приміщення після закінчення робочого дня, дотримуючись встановленого режиму огляду приміщення, необхідно:

- 1) зачинити вікна, кватирки;
- 2) перевірити приміщення та переконатися у відсутності тліючих предметів;
- 3) відключити від електромережі всі електроприлади,

електрообладнання та вимкнути освітлення.

4.3 ВИСНОВКИ

Виходячи з проведеної оцінки робочого місця програміста можна сказати, що воно задовольняє умовам, що повинні забезпечувати комфортну роботу. Розміри розглянутого приміщення відповідають нормативним значенням площі та об'єму з розрахунку на одне робоче місце. Оскільки, за звичайних умов, влітку у приміщенні досить жарко, а взимку холодно, для створення й автоматичної підтримки оптимальних значень температури, вологості, чистоти і швидкості руху повітря незалежно від зовнішніх умов, у холодний час року використовується водяне опалення, у теплий час року застосовується кондиціонер. Незважаючи на наявність останнього, сумарний рівень шуму задовольняє вимогам, що встановлені для програмістів ЕОМ, і додаткові заходи для зниження рівня шуму в приміщенні не потребуються. Так як природного освітлення в розглянутому приміщенні недостатньо для нормальної роботи, то використовується штучне освітлення, що задовольняє норму освітленості у 300-500 лк згідно з ДБН В.2.5-28-2006. Також були зазначені розміри робочого столу і крісла, параметри розташування екрану монітору відносно програміста, які забезпечували б комфортну роботу.

Дотримання умов, що визначають оптимальну організацію робочого місця програміста, дозволить зберегти працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному значенні продуктивність праці програміста. Також в процесі роботи з комп'ютером необхідно дотримуватися правильного режиму праці і відпочинку, оскільки в протилежному випадку можливі перенапруга зорового апарату, головні болі, дратівливість, порушення сну, стомленість і больові відчуття в області шиї і кистях рук.

ВИСНОВКИ

В даній роботі було розглянуто таку тему як композитні семантичні веб-додатки у наукових дослідженнях. Були роз'ясненні такі поняття як композитні веб-сервіси(мешапи), семантичний веб, онтології, бази знань, технології реалізації семантичних додатків. Під час пошуку інформації та розуміння концепції семантичних веб-додатків було створено власну програму з метою ознайомлення з принципами, методиками і технологіями розробки таких веб-додатків. Було реалізовано приклади використання декількох точок доступу SPARQL для отримання інформації з онтологій та RDF-файлів в залежності від запитів SPARQL. Були використані бібліотеки для роботи з SPARQL запитами і RDF-файлами. Практична реалізація семантичного додатку забезпечило краще розуміння необхідності використання такої концепції побудови інформації. Такий підхід організації даних та доступу до них дає більше можливостей ніж поширена зараз конструкція СУБД + інтерфейс.

Розвиток технологій семантичних Веб-сервісів дасть можливість створювати єдине уніфіковане представлення семантичного Веб-сервісу в різних застосуваннях, дозволить точно знаходити необхідну інформацію, спрощувати корпоративну інтеграцію та інтеграцію мережевих додатків в розподіленому Веб-просторі. Семантика забезпечує автоматизацію динамічного пошуку, композиції і виконання Веб-сервісів. Акценти зміщуються від синтаксичної до семантичної інтероперабельності на основі підходів прийняття рішень.

Семантичний Web обіцяє цілком відчутні переваги, додаткові сервіси. Навігація в Мережі стане більш осмисленою, а пошук - більш точним. Самі користувачі зможуть створювати сторінки Семантичного Web, давати власні визначення і вводити нові правила виведення, використовуючи стандартне для цієї мережі програмне забезпечення.

В даний час на Семантичний Web працює безліч наукових підрозділів по всьому світу, удосконалюючи і розробляючи нові протоколи, технології,

середовища програмування, агенти, мови, інтерфейси, методи розподіленого пошуку знань. Прогнозується, що працездатна глобальна версія Семантичної мережі з'явиться вже в цьому десятилітті. Про реальність цього прогнозу свідчить публікація та затвердження WWW-консорціумом у лютому 2004 року фінальних версій двох основних специфікацій Семантичного Web. Це переглянута версія RDF (до неї додані опису тестів, що дозволяють додаткам на різних мовах програмування розуміти один одного, а також засоби стикування RDF і XML) і OWL.

Можна виділити наступні переваги застосування семантичних композитних веб-додатків у наукових дослідженнях:

- Системи пошуку наукових даних “за смислом”, а не за ключовими поняттями. “Розумний пошук” по базах експериментальних даних, базах публікацій, термінів понять, законів, теорем і т.д.
- Можливість побудови автоматизованого циклічного процесу здобуття нових знань з існуючих в наукових дослідженнях.
- Спрощення інтеграції наукових мережевих додатків та баз знань в розподіленому Веб-просторі. Створення спеціалізованих композитних додатків з існуючих, семантично описаних мережевих компонентів.

У підсумку, такого роду семантичний додаток і одержувана його засобами інформація відкривають нові можливості для розвитку наукових досліджень, забезпечують нові технології для наукової та науково-організаційної діяльності.[15]

ПЕРЕЛІК ПОСИЛАНЬ

1. Семантический веб: от идеи - к технологии. Дмитрий Ланде [Електронний ресурс]. — Режим доступу : <http://poiskbook.kiev.ua/sw.html>. — Дата доступу : травень 2015.
2. Веб 2.0 та Семантичний Веб [Електронний ресурс]. — Режим доступу : <http://it.ridne.net/web2-overview>. — Дата доступу : травень 2015.
3. В. Деречкий Подходы и задачи композиции сервисов [Текст] / В. Деречкий // Проблемы програмування. — 2008. — №4 — С. 478.
4. Ф. Андон, В. Деречкий Роль семантики в интеграции приложений на основе веб-сервисов [Текст] / Ф. Андон, В. Деречкий // . — 2009. — С. 478.
5. Модель-вид-контролер [Електронний ресурс]. — Режим доступу : <http://uk.wikipedia.org/wiki/Модель-вид-контролер>. — Дата доступу : травень 2015.
6. Пример последовательного построения SPARQL-запроса [Електронний ресурс]. — Режим доступу : http://test.wikivote.ru/index.php/Пример_последовательного_построения_SPARQL-запроса. — Дата доступу : травень 2015.
7. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98 [Електронний ресурс]. — Режим доступу : <http://mozdocs.kiev.ua/view.php?id=2445>. — Дата доступу : травень 2015.
8. Гігієнічна класифікація праці за показниками шкідливості та небезпечності факторів виробничого середовища, важкості та напруженості трудового процесу [Електронний ресурс]. — Режим доступу : <http://zakon2.rada.gov.ua/laws/show/v4137400-86>. — Дата доступу : травень 2015.

9. Санітарні норми мікроклімату виробничих приміщень. ДСН 3.3.6.042-99 [Електронний ресурс]. — Режим доступу : <http://mozdocs.kiev.ua/view.php?id=1972>. — Дата доступу : травень 2015.

10. Санітарні норми виробничого шуму, ультразвуку та інфразвуку. ДСН 3.3.6.037-99 [Електронний ресурс]. — Режим доступу : <http://mozdocs.kiev.ua/view.php?id=1789>. — Дата доступу : травень 2015.

11. Державні будівельні норми України. ДБН В.2.5-28-2006. Природне і штучне освітлення [Електронний ресурс]. — Режим доступу : <http://diamantsklo.com.ua/news/architek/dbn25282006>. — Дата доступу : травень 2015.

12. Правила охорони праці під час експлуатації електронно-обчислювальних машин. НПАОП 0.00-1.28-10 [Електронний ресурс]. — Режим доступу : <http://document.ua/pravila-ohoroni-praci-pid-chas-ekspluatatsiyi-elektronno-obch-nor17970.html>. — Дата доступу : травень 2015.

13. Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою. НАПБ Б.03.002-2007 [Електронний ресурс]. — Режим доступу : <http://document.ua/normi-viznachennja-kategorii-primishen-budinkiv-ta-zovnishni-nor7322.html>. — Дата доступу : травень 2015.

14. Пожежна безпека на об'єктах будівництва ДБН В.1.1.7–2002 [Електронний ресурс]. — Режим доступу : <http://document.ua/zahist-vid-rozhezhi.-rozhezhna-bezpeka-obektiv-budivnictva-nor300.html>. — Дата доступу : травень 2015.

15. Вініченко, А.М. Семантичні композитні веб-додатки в наукових дослідженнях [Текст] / А.М. Вініченко // Системний аналіз та інформаційні технології: матеріали 17-ї Міжнародної науково-технічної конференції SAIT 2015, Київ, 22-25 червня 2015 р. – 2015. – с.148