дріжджових клітин, може бути рекомендований як основна сировина в технології спирту етилового.

Список використаних джерел:

1. Гідроліз крохмалю в соку цукрового сорго / Н. О. Григоренко, Л. Г. Білостоцький, Н. І. Штангеєва та ін. // Харчова і переробна промисловість. – 2007. – № 5(33). – С. 29-30.

2. Григоренко Н. О. Перспективи технологічного перероблення цукрового сорго / Н. О. Григоренко, Л. Г. Білостоцький, Н. І. Штангеєва.

3. Ковальчук, В. П. Цукрове сорго – цукровмісна сировина та потенційне джерело енергії / Ковальчук В. П., Григоренко Н. О., Костенко О. І. // Цукрові буряки. – 2009. – № 6. – С. 6-7.

4. Панова Т. М., Щеголев А. А. Технология и оборудование для переработки растительного сырья. Методические указания по выполнению лабораторного практикума для студентов очнов и заочной форм обучения. – Екатеринбург. 2010. – 17 с.

5. Соргові культури: технологія, використання, гібриди та сорти / А. В. Черненков, М. С. Шевченко, Б. В. Дзюбенський та ін. // Дніпропетровськ. – 2011. – 64 с.

Либа К.О.

студентка,
Національний технічний університет України
«Київський політехнічний інститут»

СЕМАНТИЧНИЙ ВЕБ З JAVASCRIPT

На сьогоднішній день комп’ютери є невід’ємною частиною нашого життя, але, нажаль, їх функції зводяться до збереження, пошуку та відображення інформації. У той час як за створення, оцінку, класифікацію та актуалізацію даних до сих пір відповідає людина. Інформація, яку зберігає або шукає машина, не має для комп’ютера ніякого значення. Це лише набір нулів та одиниць. Але що можна зробити, щоб змінити це? Якщо комп’ютер не може розуміти людської мови, то потрібно використовувати мову, яку б він зрозумів. Тобто, вся інформація, що розміщується в Інтернеті, повинна бути написана двома мовами: для комп’ютера і для людини. Семантична павутина – це концепція мережі, у якій кожний ресурс, що згадується в Інтернеті, має бути доповнений описом, зрозумілим комп’ютеру. Як казав Тім Бернес Лі, «Семантична павутина дозволить машинам розуміти семантику документів і даних, але не людських промов і записів».

Хоча Семантичний Веб очолюють такі гіганти, як Тім Бернес Лі, до сих пір розробницька спільнота не поспішає впроваджувати його у реальному житті. Перша специфікація для RDF – Resource Description Framework, ключовий документ для опису ресурсів семантичної мережі, з’явилась у 1997 році, у 2004 було затверджено остаточну рекомендацію для RDFS – основа для опису онтологій з використанням розширеного RDF-словника, у 2004 (у 2009 році OWL 2) році з’явилася мова опису веб-ресурсів – Web Ontology Language

(OWL), а також у 2008 році – мова запитів і протокол передачі даних, описаних за допомогою RDF-класів – SPARQL. Але, на привеликий жаль, лише останнім часом використання семантичних технологій починає набирати обертів.

Наразі існує широкий спектр інструментів, доступних для розробників, які хочуть підключитися до Семантичної мережі. Більшість мов програмування високого рівня пропонують бібліотеки, такі як Jena для Java, RDFLib для Python і Ruby, Redland інтерфейс для Ruby. Але, здається, не існує бібліотеки, доступної для Javascript, що запропонувала б можливості роботи із Семантичним Веб.

Javascript – це мова програмування, яка була створена у 1995 році Бренданом Ейком для внутрішнього користування компаніїєю Netscape. Вона швидко отримала популярність як мова програмування, яку дуже легко включити до веб-сторінки та почати використовувати, але довгий час не була признаною спільнотою як самостійна мова програмування. Так, перші специфікації мали багато помилок, недосконалий дизайн, недостатня кількість книжок для вивчення мови, її навіть називали мовою для аматорів. Але з часом помилки було виправлено, книжки написано, специфікаціїї вдосконалились і Javascript наразі вважається самостійно мовою. З кожним роком її популярність все зростає, все більше і більше з'являється бібліотек. Нещодавно Javascript почав використовуватися не лише у браузерах, а і для розробки серверної частини додатків. Вбачаючи такий прогрес та збільшення спільноти, я вважаю, що Javascript заслуговує отримати можливість створювати додатки, що стануть частиною настільки очікуваного Семантичного Веб.

Архітуктура семантичного додатку зображена на рис. 1.

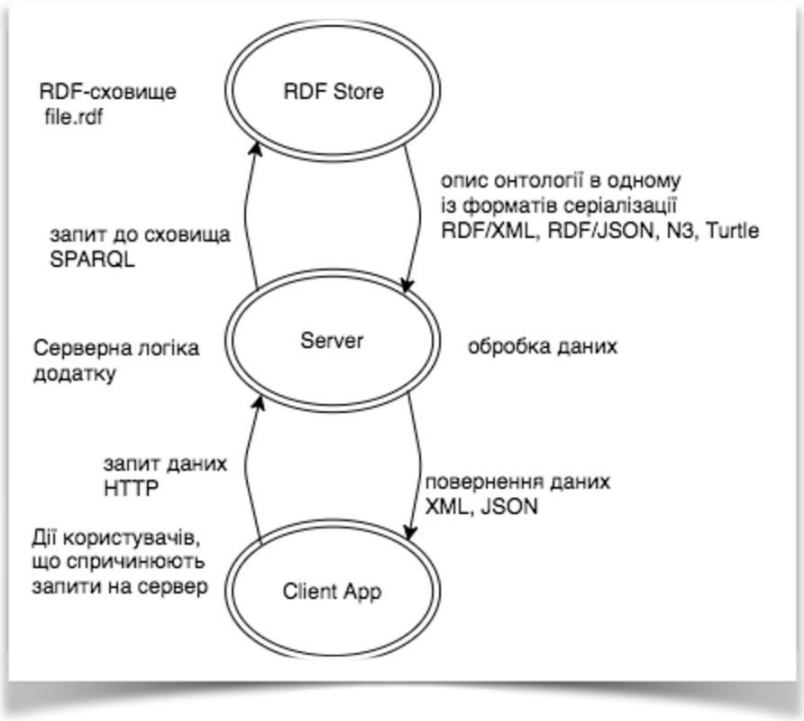

Рис. 1. Архітектура семантичного веб-додатку

Ми маємо RDF-сховище – в найпростішому випадку це файл з розширенням.rdf що містить описи онтологій і являє собою базу знань. Для запиту даних із бази знань будемо використовувати мову запитів SPARQL. Також ми маємо сервер – для javascript це додаток який виконується у середовищі Node.js, який буде отримувати http запити від Client App – клієнтського додатку, робити запит до RDF-сховища, отримувати дані в одному з форматів серіалізації – RDF/XML, RDF/JSON, N-Triples, N3, Turtle або RDFa та повернути відповідь. Задачею сервера є обробити ці дані та повернути їх у зрозумілому клієнту форматі. Клієнт, у свою чергу, обробить результат та покаже оновлене представлення користувачеві. Але, як відомо, SPARQL надає тільки методи для отримання даних, які присутні у базі знань, а також не надає методу для оновлення даних, тому що вважається, що всі ресурси у семантичній мережі є публічними, і тому надання можливості редагування даних кожному охочому є безглуздим. Саме тому у нашому додатку сервер також буде відповідальний за додання та зміну онтологій вручну у нашій базі знань.

Проаналізувавши всі існуючі рішення для реалізації семантичного веб на мові Javascript, я зупинилася на використанні сховища N3.js – бібліотека для роботи з RDF-базою знань, що була розроблена Рубеном Верборхом та надає можливість зчитувати дані з RDF-сховища у N3 сумісному форматі, та представляти їх у вигляді об'єкта-триплета: { subject, predicate, object}, також створювати нові онтології в тому ж вигляді. Ще однією перевагою є функціональність для ініціалізації сховища у пам'яті у вигляді графа, що надає можливість швидкого доступу до даних як для зчитування, так і для запису, що обумовлено особливістю реалізації мови Javascript під назвою Fast Property Access.

Для розробки серверу я обрала фреймворк під назвою Express, що дозволяє писати серверний код, що буде виконуватись у середовищі Node.js. Це середовище пропонує нам ряд «доступних із коробки» модулів, таких як http – для роботи з http запитами, або fs – file system, для роботи із файловою системою, а саме нас цікавить доступ до статичних файлів [6].

Далі, нам потрібний фрейморк для написання клієнтської частини. Я звернула увагу на так званий «new generation web framework» під назвою Aurelia, який використовує найновіші технології, а саме нову специфікацію EcmaScript 6 та систему контролю модулів jspm, реалізовує патерн MVVM, пропонує «two-way data-binding» та має сучасну архітектуру. Так як більшість браузерів ще не підримують можливості ES6, тому Aurelia використовує бібліотеку Babel для транспіляції коду ES6 у код ES5 [7].

В цій статті було зроблено огляд на сучасний стан Семантичного веб у мастшабі Javascript. Базуючись на даних, отриманих під час аналізу існуючих інструментів, було розроблено архітектуру семантичного додатку та обрано бібліотеки для написання семантичного веб-порталу. Але це лише перші кроки, основну частину дослідження буде зроблено під час розробки додатку в цілому, а також Javascrupt-бібліотеки, яка надаватиме можливість простого доступу до RDF-даних, включаючи створення нових онтологій.

Список використаних джерел:

1. Berners-Lee T., Hendler J., Lassila O. The Semantic Web // Scientific American 2001. – May. – P. 598–602.
2. Офіційний сайт: SPARQL Query Language for RDF. – Режим доступу: http://www.w3.org/TR/rdf-sparql-query/. – Электронна публікація. – Дата доступу: 14.05.2016.
3. Офіційний сайт: RDF 1.1 Concepts and Abstract Syntax. – Режим доступу: https://www.w3.org/TR/rdf11-concepts/ – Электронна публікація. – Дата доступу: 14.05.2016.
4. Офіційний сайт: OWL 2 Web Ontology Language. – Режим доступу: http://www.w3.org/TR/owl2-overview/ – Электронна публікація. – Дата доступу: 14.05.2016.
5. Semantic tools, KADS22 Інтернет–ресурс. – Режим доступа: http://www.semantictools.ru/tools/12––sw.html – Дата доступу: 14.05.2016.
6. Офіційний сайт: NodeJS. – Режим доступу: http://nodejs.org – Дата доступу: 20.05.2016.
7. Офіційний сайт: AureliaIO. – Режим доступу: http://aurelia.io/ – Дата доступу: 20.05.2016.

Масюк А.С.

аспірант;

Левицький В.Є.

доктор технічних наук, професор,
Національний університет «Львівська політехніка»

ПОЛІАМІДНІ І ПОЛІПРОПІЛЕНОВІ КОМПОЗИТИ З ПОЛІМЕР-СИЛІКАТНИМ НАПОВНЮВАЧЕМ: МОРФОЛОГІЯ І ВЛАСТИВОСТІ

Сьогодні створення нових полімерних композиційних матеріалів (ПКМ) на основі термопластичних матриць і неорганічних наповнювачів є однією з найбільш перспективних областей технології полімерних і композиційних матеріалів, зокрема розроблення композитів з підвищеними технологічними та експлуатаційними характеристиками. У цей же час, серед широкого кола термопластичних полімерів конструкційного призначення значного поширення набули полікапроамід (ПА-6) та поліпропілен (ПП).

Поряд з цим, підвищення технологічних і експлуатаційних властивостей ПКМ переважно досягається у випадку, коли наповнювач має підвищену технологічну сумісність до полімерної матриці і зберігаються міжмолекулярні і міжфазні взаємодії між інгредієнтами композиту під час експлуатації виробів на його основі. Основною проблемою, що виникає під час створення ПКМ, є низька сумісність наповнювача і полімерної матриці. Серед методів підвищення сумісності між компонентами ПКМ широко використовуються методи, пов’язані з попереднім модифікуванням неорганічних наповнювачів функційноактивними високомолекулярними сполуками.

Для наповнення ПА-6 «Гроднамид 210/310» і ПП «Moplen» використовували Ni-вмісний полімер-силікатний матеріал (Ni-ПСМ), одержаний сумісним осадженням натрієвого рідкого скла та полімерних