

Слухай Яна Олександрівна

магістр комп'ютерних наук

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

Слухай Яна Александровна

магистр компьютерных наук

Национального технического университета Украины

«Киевский политехнический институт имени Игоря Сикорского»

Slukhai Yana

Master of Computer Science of the

National Technical University of Ukraine

«Igor Sikorsky Kyiv Polytechnic Institute»

ДОСЛІДЖЕННЯ АРХІТЕКТУРНИХ ВИМОГ ДО БАЗИ ЗНАНЬ ЯК СЕРВІСУ

ИССЛЕДОВАНИЕ АРХИТЕКТУРНЫХ ТРЕБОВАНИЙ К БАЗЕ ЗНАНИЙ КАК СЕРВИСУ

THE KNOWLEDGE BASE AS A SERVICE ARCHITECTURAL REQUIREMENTS RESEARCH

Анотація. Досліджено основні вимоги до побудови бази знань як сервісу, проаналізовано переваги використання REST-архітектури для такої системи.

Ключові слова: веб-сервіс, семантичний веб, REST, сховище триплетів, RDF-сховище.

Аннотация. Исследованы основные требования к построению базы знаний как сервису, проанализированы преимущества использования REST-архитектуры для такой системы.

Ключевые слова: веб-сервис, семантический веб, REST, хранилище триплетов, RDF-хранилище.

Summary. The basic requirements for building a knowledge base as a service have been researched, the advantages of using REST-architecture for such a system have been analyzed.

Key words: web-service, semantic web, REST, triplestore, RDF store.

Все більш цікавими та різноманітними стають розробки у сфері автоматичного пошуку, збору і обробки інформації. Одним з провідних напрямків цієї галузі є сукупність технологій семантичного вебу. Побудова масштабних, кросплатформних систем, які б дозволяли використовувати всі переваги семантичного вебу та сервісної архітектури є першочерговою задачею для дослідників пов'язаних даних. Ключевим поняттям цього процесу стає онтологія.

У загальних рисах під онтологією розуміється система понять деякої предметної області, яка представляється як набір сутностей, з'єднаних відношеннями. Онтології використовуються для формальної специфікації понять і відношень, які характери-

зують певну область знань. Перевагою онтологій як способу представлення знань є формалізована структура, яка спрощує комп'ютерну обробку [1].

У явному вигляді онтології використовуються як джерела даних для багатьох комп'ютерних програм (для інформаційного пошуку, аналізу текстів, вилучення знань та інших інформаційних процесів), дозволяючи більш ефективно обробляти складну і різноманітну інформацію [3, с. 37–38]. Цей спосіб представлення знань дозволяє додатком розпізнавати ті семантичні відмінності, які є зрозумілими для людей, але не є відомими для комп'ютера. Повноцінна розробка онтологій в новому розумінні цього терміну почалася лише в кінці 90-х. Це досить нова і мало досліджена галузь прикладної лінгвістики.

RDF (Resource Description Framework) специфікація передбачає створення деякої множини ресурсів, для яких визначаються зв'язки [1]. RDF семантика має визначати онтологію конкретної предметної області. Саме онтології з кожним роком завойовують все більше застосування у вирішенні задач представлення знань, семантичного представлення інформаційних ресурсів та їх пошуку.

Ресурсом в RDF може виступати будь-яка сутність — як інформаційна (наприклад, веб-ресурс чи зображення), так і неінформаційна (наприклад, тварина, вулиця чи якесь інше нематеріальне поняття). Ствердження, висловлене про ресурс, має вигляд «суб'єкт — предикат — об'єкт» і називається триплетом [2, с. 136–137]. Твердження «небо голубого кольору» в RDF-термінології можна представити наступним чином: суб'єкт — «небо», предикат — «має колір», об'єкт — «голубий». Для позначення суб'єктів, відношень і об'єктів у RDF використовуються URI [4, с. 44–45].

Все більш популярною стає сервісно-орієнтована архітектура (Service-oriented architecture, SOA) — архітектурний шаблон програмного забезпечення, модульний підхід до розробки програмного забезпечення, заснований на використанні розподілених, слабко пов'язаних замісних компонентів, оснащених стандартизованими інтерфейсами для взаємодії за стандартизованими протоколами.

Ідеї компонування сервісно-орієнтованої архітектури та семантичних сховищ є досить привабливими завдяки наданню користувачу досить зручного способу маніпулювання даними. Такі проекти зазвичай реалізуються за допомогою сховищ триплетів, до яких надається доступ згідно протоколів передачі даних, наприклад за допомогою HTTP.

Сховище триплетів (RDF-сховище) — це тип графічної бази, що зберігає семантичні факти. RDF, що означає Resource Description Framework, є моделлю для публікації та обміну даними в Інтернеті, стандартизований W3C [3, с. 39].

Сховища триплетів є більш гнучкими та менш дорогими, ніж, наприклад, реляційна база даних. RDF-сховище також може бути здатним обробляти потужні семантичні запити та використовувати логіку виведення для виявлення нової інформації поза існуючими відношеннями.

Важливим є вибір архітектурного стилю для проектування бази знань як сервісу. Порівнявши найбільш часто використовувані архітектури SOAP (Simple Object Access Protocol) та REST (Representational State Transfer), було визначено, що REST краще підходить для вирішення таких задач. Серед його переваг над SOAP можна виділити наступні:

- SOAP — це ціле сімейство протоколів і стандартів, звідки напряму випливає, що це більш важ-

кий і складний варіант з точки зору машинного оброблення. Тому REST працює швидше.

- SOAP використовує HTTP як транспортний протокол, в той час як REST базується на ньому. Це означає, що всі існуючі напрацювання на базі протоколу HTTP, такі як кешування на рівні сервера, масштабування, продовжують працювати також в REST-архітектурі, а для SOAP необхідно шукати інші засоби.
- Відповідь REST може бути представлена в різних форматах, а SOAP прив'язаний до XML [2, с. 135].
- SOAP працює з операціями, а REST — з ресурсами.
- Всі запити у REST можна поділити на 4 типи у відповідності з CRUD, причому кожен тип відповідає HTTP методу — Post, Get, Put, Delete
- Всі ресурси у REST мають унікальний ідентифікатор (URL)
- Всі операції клієнта з сервером у REST не використовують збереження стану, тобто сервер не повинен зберігати взагалі ніякої інформації про клієнта [5, с. 114–115].

Також цікавим є розгортання такої системи у хмарі, адже при використанні хмарних обчислень програмне забезпечення надається користувачеві як Інтернет-сервіс. Користувач має доступ до власних даних, але не може управляти і не повинен піклуватися про інфраструктуру, операційну систему і програмне забезпечення, з яким він працює.

Прикладні інформаційні системи можуть взаємодіяти з RDF-сховищем за допомогою спеціального API або структурованих SPARQL-запитів. SPARQL — це стандартна мова запитів до RDF-даних. Для найбільш ефективної роботи з базою знань за допомогою сервісу можна виділити такі вимоги:

1. Можливість виконання SPARQL-запитів
2. Наявність REST-API з реалізацією доступу до ресурсів, які найбільш часто будуть потрібні користувачам (тут важливою є постановка вимог до системи кінцевими користувачами для визначення таких ресурсів)
3. Можливість розгортання системи у хмарі (для забезпечення кращої масштабовності та універсальності)
4. Кросплатформність (дозволяє суттєво скоротити витрати на розробку нового або адаптацію існуючого програмного забезпечення).

Визначення вимог до бази знань як сервісу може значно скоротити час на вибір програмного забезпечення, бібліотек та фреймворків для її реалізації. Це також допоможе уникнути проблем при необхідності масштабувати чи розгорнути систему у різноманітних (у тому числі хмарних) середовищах. Отже, дане дослідження є підґрунтям для подальшої реалізації бази знань як сервісу.

Література

1. Guarino N. Understanding, Building, and Using Ontologies / N. Guarino // [Электронный ресурс] Режим доступа: <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/guarino/guarino.html> — 01.02.2018
2. Fielding, R. (2000). Architectural Styles and the Design of Network-based Software Architectures (Ph.D.) / R. Fielding // University of California, Irvine f- C. 132–139.
3. Hayes-Roth, Frederick, Waterman D., Douglas L. (1983). Building Expert Systems / F. Hayes-Roth, D. Waterman, L. Douglas // Addison-Wesley. ISBN0–201–10686–8. — С. 38–39.
4. Hayes, P., Horrocks, I., Patel-Schneider, P: OWL Web Ontology Language Semantics and Abstract Syntax / P. Hayes, I. Horrocks, P. Patel-Schneider // W3C Recommendation, 10 February 2004. — С. 44–45.
5. Richardson L., Ruby S. (2007), RESTful Web service / L. Richardson, S. Ruby // O'Reilly Media, ISBN978-0-596-52926-0. — 114–115 с.