

**УДК 004.4'2**

**Л.В.Корчук**

Тернопільський національний технічний університет імені Івана Пулюя, Україна

## **СУМІСНІ СТАНДАРТИ ТА МОВИ РОЗМІТКИ RDF**

**L.V.Korchuk**

### **SUPPORTED STANDARDS AND RDF MARKUP LANGUAGE**

RDF працює з XHTML, який на відміну від звичайного HTML більш розширений. Також використовується підключення однієї з бібліотек, що дозволяє розпізнавати префікс (dc), який використовується для розпізнання метаданих RDF. Не можна просто написати title і creator, оскільки в XHTML не зарезервовано таких ключових слів. Тому і був визначений деякий префікс і ключові слова, щоб пошуковий робот відразу визначав, що це саме RDF. Далі він дивиться ключове слово і бере його значення. Тепер робот знає, що наприклад Заголовок статті – це саме заголовок статті, а не якийсь текст, випадково вкладений в тег <h2>.

RDF в XHTML використовує багато XHTML-атрибутів, а також надає кілька нових. Атрибути, вже наявні в XHTML, будуть мати те ж значення, що і в XHTML, хоча їх синтаксис може бути дещо змінений. В XHTML не чітко визначений спосіб додавання нових значень; RDF вирішує цю проблему, дозволяючи використовувати URIs в якості значень. Він також вводить ідею 'компактного URI', що дозволяє коротко висловлювати повне значення URI.

Структуровані дані, до яких RDF надає доступ, представляють собою колекцію операторів. Оператор – це базова одиниця інформації, сконструйована в певному форматі для полегшення обробки. У свою чергу, розбиваючи великі набори інформації на колекції операторів, навіть дуже складні метадані можна обробляти з використанням простих правил [3].

Щоб зробити цю інформацію доступною для машинної інтерпретації, RDF визначає структуру операторів. Оператор формально називається триплет, тобто він складається з трьох компонент. Перший – це суб'єкт триплету, те з чим працюють наші оператори. Друга частина триплету – властивості суб'єкту, які ми хочемо визначити. Зазвичай в RDF їх називають предикатами. Остання частина триплету називається об'єктом.

Розбивка оброблюваної інформації на зручні для обробки блоки дозволяє специфікувати наші дані, але тут є деяка двозначність. RDF вирішує цю проблему, замінюючи наші невизначені терміни на URI-посилання.

URIs найчастіше використовуються для ідентифікації веб-сторінок, але RDF застосовує їх для надання унікальних ідентифікаторів понять. Наприклад, ми можемо ідентифікувати суб'єкт всіх наших операторів (перша частина кожного триплету), використовуючи DBPedia URI [2].

URI-посилання також використовуються для унікальної ідентифікації об'єктів в операторах метаданих (третя частина кожного триплету). Фото – це вже URI, але ми можемо також використовувати URI для унікальної ідентифікації країни. У той же час ми можемо вказати, що ім'я і дата народження в дійсності є літералами (а не URIs), помістивши їх в подвійні лапки. URI-посилання також використовуються для гарантованої однозначності предикатів. Хоча URI-ресурси завжди використовуються для суб'єктів і предикатів, частина об'єкт триплету може бути і URI.

Деякі літерали, такі як дати і числа, мають особливі значення, тому в RDF є механізм для вказування типу літералу, вказується шляхом приєднання URI в кінець, і цей URI вказує тип даних літералу.

У самому RDF немає єдиного способу для вираження триплетів, оскільки ключові ідеї RDF – це триплет і використання URIs, а не конкретний синтаксис. Однак є кілька механізмів для вираження триплетів: це RDF / XML, Turtle і, звичайно, RDF. RDF

використовує Turtle-синтаксис для роз'яснення своїх ідей, оскільки він досить компактний. Колекція триплетів називається graph [1, 5].

Для створення компактного вираження RDF-операторів RDF дозволяє скорочення всіх URI посилок у форму, названу 'компактний URI' або CURIE. Використання впроваджених метаданих полягає в запозиченні фрагментів розмітки і переміщенні їх з одного документа в інший. Це може відбуватися при використанні таких утиліт, як drag-and-drop в браузері, або через ділянки коду, що надаються авторами для включення в їхні документи. Дана специфікація не вказує, як фрагменти XHTML+RDF повинні оброблятися, коли вони знаходяться поза повним XHTML+RDF-документом.

До всього цього RDF ще й підтримує Schema.org, так як використовується в ній модель даних дуже універсальна і є похідною від схеми RDF [4].

Schema.org – це стандарт семантичної розмітки даних в мережі, оголошений пошуковими системами Google, Bing і Yahoo! влітку 2011 року. Мета семантичної розмітки - зробити Інтернет більш зрозумілим, структурованим і полегшити пошуковим системам і спеціальним програмам вилучення та обробку інформації для зручного її подання в результатах пошуку. Багато сайтів генеруються із структурованих даних, які найчастіше зберігаються в БД. Коли дані переводяться в HTML, стає дуже складно відновити початкові структуровані дані. Багато додатків, особливо пошукові системи, можуть отримати значні переваги, маючи прямий доступ до структурованих даних. Розмітка сторінок дозволяє пошуковим системам розуміти інформацію на веб-сторінках і надавати користувачеві саме ту інформацію, яку він шукає, також може задіяти нові інструменти і додатки, які використовують цю структуру. Розмітка відбувається безпосередньо в HTML-коді та XHTML-коді сторінок з допомогою спеціальних атрибутів і не вимагає створення окремих експортних файлів [6].

RDF іноді працює аналогічно мікроформатам. У той час як мікроформати специфікують синтаксис для впровадження структурованих даних в HTML-документи і словник специфічних термінів для кожного мікроформату, RDF специфікує тільки синтаксис і ґрунтується на незалежній специфікації термінів (таксономією) інших. RDF дозволяє вільно змішувати безліч незалежно розроблених словників і сконструйована таким чином, щоб мова могла бути розібрана без знання специфічного використовуваного словника термінів [3].

#### **Література.**

1. David Wood. Linking Government Data / David Wood – New York: Springer, 2011. – 232 p. – ( ISBN: 9781461417668).
2. Dean Allemang. Semantic Web for the Working Ontologist (2nd ed.) / Dean Allemang, Jim Hendler – Waltham: Morgan Kaufmann, 2008. – 354 p. – (ISBN: 0123859654).
3. Liyang Yu. A Developer's Guide to the Semantic Web / Liyang Yu – New York : Springer, 2011. – 608 p. – (ISBN: 9783642159695).
4. Mark Watson. Practical Semantic Web and Linked Data Applications, Java Edition / Mark Watson – Raleigh: Lulu.com, 2010 – 180 p.
5. Vipul Kashyap. The Semantic Web / Vipul Kashyap, Christoph Bussler, Matthew Moran – Berlin: Springer, 2008. – 414 p. – (ISBN: 9783540764519).
6. rucschema.org - проект перевода schema.org. Сайт коллекции схем [Электронный ресурс]. – Режим доступа: <http://rucschema.org/>. – Дата доступа: ноябрь 2016 г.