Інформаційні технології

УДК 004.65

Булах Богдан Вікторович
кандидат технічних наук,
доцент кафедри системного проектування
Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"

ОНТОЛОГІЯ ЯК АЛЬТЕРНАТИВА РЕЛЯЦІЙНІЙ БАЗІ ДАНИХ У ПРИКЛАДНИХ ПРОГРАМНИХ СИСТЕМАХ

Анотація. *Досліджено проблему вибору класу моделі збереження даних у програмних системах (онтологія чи реляційна модель). Узагальнено перелік відмінних та спільних рис цих двох моделей та наведено загальні критерії вибору моделі.*

Ключові слова: *реляційна база даних, схема даних, онтологія.*

Постановка задачі. Реляційні бази даних (далі - реляційні БД або РБД) поки залишаються найпоширенішим підходом до організації системи збереження даних для програмних продуктів. За тривалий період використання ця зріла технологія зміцнилася чималою кількістю інструментів, програмних бібліотек, реалізацій серверів БД та ін. Нині позиції РБД дещо розхитують не-реляційні БД (т.зв. NoSQL-бази даних), однак існує ще більш радикальна альтернатива - онтології та використання баз знань (БЗ) на їх основі замість РБД. Дана стаття присвячена аналізу спільних та відмінних рис РБД та онтологій в світлі їх використання у сучасних програмних продуктах.

База знань на основі онтології

Під онтологією в комп'ютерних науках, в загальному сенсі, розуміють формальний опис певної області знань з використанням класів (понять), їх зв'язків, аксіом, правил. Реалізувати програмну систему з використанням онтології означає: а) розробити модель знань, описати поняття та їх взаємозалежності (т.зв. ABox), б) обрати сховище для зберігання знань (як правило у формі триплетів "суб'єкт, предикат, об'єкт") та наповнити базу фактів з використанням доступних понять (т.зв. TBox), в) організувати доступ до бази знань (ABox + TBox) з програмного коду. Розберемо дещо детальніше спільні та відмінні риси РБД та онтологій.

Відкрита та закрита моделі світу

Однією з основних відмінностей між онтологією та РБД є трактування "відкритості світу": при роботі з онтологіями приймають відкритість світу (тобто, не можна виключати наявність будь-яких фактів, що явно не заборонені чи виключені в онтології), при роботі з РБД приймають "закритість світу" в межах тих даних, які вже є в БД. Невеликий приклад. Нехай маємо такий факт, виражений триплетом: *"Богдан" - "має громадянство" - "Україна"*. На запит щодо кількості громадянств у Богдана БД відповість "1" (українське), а БЗ - "як мінімум 1" (тобто ніде не сказано, що у Богдана немає інших громадянств, окрім українського). Фактично, мова йде про те, що системи на основі онтологій є системами з неповною інформацією, а РБД-орієнтовані системи оперують (умовно) повною інформацією. Інший аспект цього зауваження стосується унікальності імен в онтологіях та РБД. У БД одна сутність може мати лише одне ім'я. В онтології сутності можуть мати багато імен [1; 2]. Нехай ми додамо додатковий факт *"будь-хто може мати лише одне громадянство"* до вищезгаданого прикладу. Тоді і БД і БЗ дадуть однакову відповідь щодо кількості громадянств у Богдана (одне). Але варто лише

додати новий "конфліктний" факт *"Богдан" - "має громаданство" - "UA"*, і для БД це буде недопустимий стан (порушення унікальності ключа), а БЗ на основі онтології зробить висновок що "Україна" та "UA" є іменами однієї сутності. Ці особливості, звичайно, слід враховувати при проектуванні логіки роботи інформаційних систем.

Методологія створення

Вважаться [1; 2], що однією з відмінностей між онтологіями та БД є те, що перші створюються на основі існуючих онтологій, в той час як БД проектують "з чистого аркуша". В реальному світі ніхто не завадить спроектувати фрагменти БД або її в цілому за конкретним існуючим аналогом. Однак сам процес створення БД та онтологій відрізняється.

Створення РБД як правило включає етапи нормалізації, чого немає в онтологіях. Коректні відношення (реляції) між таблицями неможливі без ключів (первинних, унікальних, ненульових), в той час, як в онтологіях (як вже вказувалось) кожна сутність може мати кілька імен. Тобто для РБД характерні заходи, які направлені на підтримку цілісності її "закритого світу", для онтологій і моделі "відкритого світу" ці заходи не застосовуються. Натомість деякі дослідники виділяють набір патернів при створенні онтологій [1]: структурний, синтаксичний, змістовий, презентаційний та ін. Також, як правило, розробники намагаються, по можливості, спиратися на повторне використання знань та адаптувати існуючі онтології, коли це доцільно. Важливо підкреслити, що онтологія дозволяє описувати факти без явного формулювання усіх можливих зв'язків між поняттями та екземплярами, спираючись на існування розвиненої таксономії понять (відсутньої для БД), і виведення повного набору фактів з усіма зв'язками виконується автоматично з використанням засобів здійснення логічних висновків (різонерів). Для БД автоматизація

заповнення колонок реалізується в програмному коді індивідуально для кожного випадку.

Критерії доцільності використання БД та онтологій

Сформулюємо перелік відмінних рис для онтологій та БД. Онтології фокусуються на значенні, смислі того, що зберігають - семантиці. Наявність екзпемплярів для онтологій є необов'язковим, головне - таксономія понять. Смисли моделі даних при реалізації з використанням РБД майже повністю закладаються в логіку програми, виражену на конкретній мові програмування, що ускладнює розуміння смислів моделі даних людиною та подальше розширення та підтримку. РБД фокусується на роботі з сухими даними, а не знаннями, і на перше місце стають ефективність роботи з даними (нормальні форми) та забезпечення цілісності (ключі та модель "закритого світу"). Оскільки в онтологіях будь-який новий факт потенційно може призвести до кардинальної зміни картини зв'язків між поняттями та екземплярами, то створення обгорток для програмного доступу до сутностей бази знань є, як правило, недоцільним, в той час як для РБД (зі сталою структурою) типовою є автоматична генерація коду по структурі БД та автоматичне збереження програмних об'єктів у БД та їх відновлення з БД.

Втім, важливо підкреслити й спільні риси між РБД та онтологією, зокрема - подібність схеми даних БД та онтології: обидві задають модель даних, виражену формальною мовою; обидві підтримують типізацію, властивості, обмеження (в онтологіях - аксіоми). Тож не дивно, що існує чимало робіт, які розглядають можливість створення прототипу онтології по готовій схемі БД (для більшої виразності, гнучкості, формалізації семантики, легкості розширення та підтримки) або ж, навпаки, створення схеми БД по онтології (для підвищення ефективності використання та вибору існуючих, зрілих інструментів розробки та супроводу).

Тож спираючись на вищевикладене, можна зробити такі висновки щодо доцільності використання онтологій у програмних продуктах. Онтології можуть успішно бути застосовані там, де в основі закладено достатньо складну модель даних, яка є досить динамічною, проте об'єми даних є відносно невеликими. Онтології точно не підійдуть для роботи над дуже великими обсягами даних через неефективність форми зберігання даних та відносно повільну процедуру логічного виведення, що є їх невід'ємною частиною. Однак ці негативні фактори можна зменшити певними компромісними кроками, знизивши складність моделі та виразність опису (складність для логічного виведення). Також можна запропонувати в ряді випадків поєднати використання онтологій та РБД в рамках однієї системи, коли окремі відносно статичні функції, чутливі до швидкодії, реалізуються на РБД, а інша функціональність виграє від гнучкості, яку дає онтологія, при цьому онтологія може містити знання про схему БД-компаньйона. Гібридне рішення особливо цікаве при розробці розподілених сервісно-орієнтованих систем, зокрема - мікросервісних систем, коли кожен відносно незалежний модуль (сервіс) може використовувати свою власну модель даних (РБД або онтологію) залежно від призначення та функцій [3].

Програмний інструментарій

Декларуючи право онтологій на використання у прикладних системах абсолютно різного призначення не можна оминути стороною той факт, що для РБД створено чималий арсенал програмних засобів: від численних бібліотек для різних мов програмування до потужних та надійних (MSSQL, Oracle) або ж "легких" та простих серверів або драйверів баз даних (MySQL, SQLite тощо). Говорячи про онтології, слід мати на увазі набагато менше коло інструментів та їх часто експериментальну реалізацію [4]. Стек необхідних інструментів та

технлогій складають: мови опису онтологій (діалекти мови OWL, мова опису логічних правил SWRL, мова опису триплетів RDF), сховища триплетів (RDF-сховища Sesame, Apache Rya, OpenLink Virtuoso, Mulgara, Apache Jena та ін.), засоби розробки (прив'язка до RDF-моделі під різні мови програмування, OWL API, Apache Jena API та ін.), модулі логічного виведення (різонери: FaCT++, HermiT, Pellet та ін.), засоби здійснення запитів до RDF-сховищ (мова SPARQL) (наприклад, HTTP-сервер Fuseki).

Висновки. Онтології та реляційні бази даних мають чимало спільних та відмінних рис. Історично склалося так, що РБД як потенційно ефективніші за онтології при роботи з великими масивами даних, використовуються набагато частіше і мають більше засобів для розробки та використання у програмних продуктах. Було проаналізовано специфіку роботи з онтологіями та представлено випадки, коли їх використання може бути доцільнішим за БД, а також описано базовий стек засобів, які формують серверну частину системи управління даними на основі онтологій для прикладних програмних систем. Постає потреба комплексного аналізу ефективності роботи систем на базі онтологій порівняно з РБД (враховуючи складність та специфіку моделей даних, різні за виразною потужністю види логіки, конкретні алгоритми логічного виведення), що є напрямком подальших досліджень.

Література

1. Sir, Michal & Bradac, Zdenek & Fiedler, Petr. Ontology versus Database. IFAC-PapersOnLine. 48. - 2015. - pp. 220-225.
2. Motik, Boris & Horrocks, Ian & Sattler, Ulrike. Bridging the gap between OWL and relational databases. 16th International World Wide Web Conference, WWW 2007. - pp. 807-816.
3. Булах Б.В. Мікросервісна архітектура з семантичною складовою для комплексів інженерних розрахунків / Булах Б.В., Крамар О.В. //

Системний аналіз та інформаційні технології: матеріали 18-ї міжнародної науково-технічної конференції SAIT 2016, Київ, 30 травня 2016 р. – К.: ННК "ІПСА" НТУУ "КПІ". – 2016. – с. 273.

4. David C. Faye, Olivier Curé, Guillaume Blin. A survey of RDF storage approaches. Revue Africaine de la Recherche en Informatique et Mathématiques Appliquées, INRIA. -2012. - 15. - pp.11-35.