

УДК 004.82

Глибовець М. М., Бобко О. В.

МЕТОДИ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ОНТОЛОГІЙ

Скласифіковано методи автоматичної генерації онтологій, та проаналізовано можливості поєднання різних методів автоматичної генерації онтологій для покращення ефективності систем для керування знаннями.

Ключові слова: онтології, кластерний аналіз, обробка природньої мови, аналіз формальних понять, керування знаннями.

Вступ

Розвиток комп'ютерних мереж загалом та Інтернету зокрема дав можливість накопичувати великі об'єми інформації з різних галузей людської діяльності, а також здійснювати інтеграцію гетерогенних корпоративних інформаційних

систем та обмін інформацією між прикладними програмами. Використання онтологій для представлення даних робить можливою машинну інтерпретацію і здійснення автоматизованого логічного виведення.

Дослідженню онтологій присвячено багато робіт, але лишаються відкритими і набувають

дедалі більшої актуальності задачі систематизації великої кількості різномірної інформації та автоматизації її обробки.

Мета даного дослідження – виявлення наявних методів автоматичної генерації онтологій, побудова системи критеріїв їхньої оцінки й порівняння та обрання оптимального рішення для створення засобу автоматичної генерації онтологій на основі набору текстових і/або XML-подібних документів для використання у керуванні знаннями в корпоративних інформаційних системах.

Для досягнення цієї мети вважаємо за потрібне здійснити таке: виявити та описати наявні методи генерації онтологій; розподілити процес генерації онтологій на окремі стадії, виконання яких може здійснюватись різними методами і потребувати різних підходів щодо автоматизації; побудувати систему вимог до інструментів для генерації онтологій та отриманих результатів, яких слід дотримуватись при автоматизації процесу побудови онтологій; проаналізувати обрані методи згідно зі встановленими вимогам як у цілому, так і на окремих стадіях генерації.

Спроби класифікації методів і засобів побудови онтологій здійснювали протягом останніх десятих років, проте жодна з наявних робіт не відповідає поставленому у нашому дослідженні завданню. У праці [18] наведено класифікацію програмного забезпечення для генерації онтологій згідно з шістьма категоріями (початкові елементи, початкові умови, попередня обробка інформації, метод навчання, результат і засіб оцінки результату). Однак автори не дають оцінки ступеня автоматизації цих процесів і розглядають тільки найпоширеніші методи побудови онтологій на основі певного набору текстових даних.

У [9] автори докладно аналізують побудову онтологій на основі набору текстових джерел і, зокрема, класифікують ряд програмних рішень для такої побудови. Проте недоліком цього дослідження є те, що автори розглядають тільки один тип джерел даних (текстові), а у визначенні критеріїв порівняння методів побудови онтологій розглядають лише сам процес побудови, не торкаючись питання масштабованості створених онтологій та можливого збільшення автоматизму роботи цих методів.

У студії [25] наведено класифікацію онтологій за призначенням, подано оригінальний підхід до визначення життєвого циклу онтології, а також наведено перелік методів автоматичного навчання для автоматичної побудови онтологій. Здійснено категоризацію ряду найпоширенішого програмного забезпечення для автоматизації процесу генерації онтологій. Однак цей перелік

неповний і за набором розглянутих програмних засобів, і за переліком методів, обраних для побудови класифікації.

Цікавий огляд засобів автоматичної семантичної обробки документів для документо-орієнтованого керування знаннями запропоновано в [28]. У цьому дослідженні наведено критерії порівняння як до інструменту, спрямованого на виконання такого роду задач, так і до вихідних онтологій. Недоліком праці є концентрація уваги авторів тільки на документах у форматі HTML.

Оригінальне визначення життєвого циклу онтології та його стадій, разом із класифікацією засобів автоматичної побудови онтологій дано в [7]. Однак серед недоліків цього дослідження варто назвати неповний перелік категорій методів генерації онтологій, використаних для класифікації, а також те, що добірка цих методів містить лише зразки, описані в англійській літературі.

1. Процес автоматичної побудови онтологій

Наявні методології побудови (*On-To-Knowledge*, DILIGENT, TOVE, KACTUS) зосереджено на залученні до виконання основної роботи експерта. Ми натомість пропонуємо побудову життєвого циклу онтології, який передбачає їхнє створення на основі деякого набору текстових і/або XML-подібних документів із подальшим розвитком онтології, враховуючи при цьому максимальну можливу автоматизацію цього процесу. Аналіз інструментів та їхньої відповідності методам автоматичної генерації онтологій здійснюватиметься на кожній стадії роботи з онтологією, коротку характеристику яких ми подаємо нижче.

Підготовча стадія. На цьому етапі виконується попередня обробка джерела даних: приведення до потрібного формату, усунення зайвих даних (напр. HTML-тегів), видалення стоп-слів тощо. Самі джерела даних можуть бути трьох типів: структуровані (напр. дані в таблицях), напівструктуровані (напр. XML-документи) та неструктуровані (не форматований текст).

Аналіз. За допомогою обраного методу (аналіз природної мови, аналіз формальних концептів тощо) відбувається отримання даних, необхідних для побудови онтології, з подальшим їх аналізом для виявлення концептів, екземплярів (або ж побудова їх опису чи анотації, напр. у випадку з текстовими документами) та відношень між ними. Щодо останнього слід розрізнити генерацію таксономічних відношень (типу «is-a», «subclass-of»), що в цілому легко встановлюються математичними чи лінгвістичними методами, і нетаксономічні, знаходження

та найменування яких потребує залучення експерта.

Валідація. На стадії аналізу можуть виникнути некоректно визначені концепти, а також зв'язки між ними. Таким чином, виникає потреба у перевірці згенерованої онтології на цілісність, несуперечливість, коректність визначення концептів і зв'язків та, в разі потреби, надання їй коректного вигляду. Стадія валідації зазвичай виконується експертом і, попри можливість виконання цієї стадії програмними засобами, питання її автоматизації здебільшого лишається відкритим.

Модифікація. Враховуючи можливість постійної зміни у предметній області, надбання нових джерел даних, зміни вимог програмного забезпечення, яке використовується для роботи з онтологіями, онтології також можуть зазнавати постійних модифікацій.

Стадія модифікації може відбуватися у кількох варіантах чи їхньому поєднанні:

- злиття та узгодження кількох існуючих онтологій;
- зміна структури існуючої онтології: доповнення чи зміна кількості концептів, атрибутів, екземплярів, таксономічних і нетаксономічних відношень.

Якщо наявна стадія модифікації, то за нею обов'язково відбувається повторна валідація модифікованої онтології, отриманої внаслідок зміни і/або злиття.

Серіалізація. Ця стадія передбачає збереження онтології як однієї з мов опису онтологій (RDF, OWL, DAML/OIL тощо) або ж реляційної бази даних. Вона може відбуватися довільну кількість разів упродовж певного витка життєвого циклу онтології і зазвичай не викликає проблем з автоматизацією.

2. Критерії оцінки онтологій і методів їхньої генерації

Метою побудови онтології є створення моделі, придатної для структуризації всієї релевантної інформації в деякій предметній області. Враховуючи те, що навіть для предметних областей із простою структурою неможливий експертний контроль за всім інформаційним потоком, засіб, призначений для реалізації цієї мети, повинен характеризуватись якнайбільшим ступенем автоматизації операцій на всіх етапах генерації, автономністю від зовнішніх джерел даних і застосовністю до текстової та гіпертекстової інформації з різними характеристиками. Предметна область зазвичай містить динамічну частину, відповідно – інструмент повинен мати здатність змінювати онтологію, додавати та видаляти концепти, екземпляри, змінювати відношення та пе-

ревіряти коректність подібних змін. Слід зважати на те, що межі між концептами у справжній предметній області можуть бути нечіткими. Зв'язки між ними можна іноді охарактеризувати певною лінгвістичною зміною, відповідно, бажаною є можливість вираження такого роду інформації в онтологіях.

Систему критеріїв можна умовно поділити на вимоги до інструменту та вимоги до онтології.

2.1. Вимоги до інструменту

Основним критерієм, за яким оцінюватимемо методи та засоби генерації онтологій, є рівень автоматизації процесу на кожному етапі. Якщо неможливо автоматизувати всі стадії генерації онтологій, використовуючи певний інструмент, варто зазначити, які саме стадії можливо автоматизувати. Автоматизація передбачає відповідність самого інструменту й згенерованої ним онтології іншим критеріям, зокрема, розширюваності (масштабованості), цілісності й точності.

Розвиток чи будь-які зміни стану предметної області потребують відображення в онтології. Тому остання має адекватно задавати динамічну характеристику предметної області. Відповідно, інструмент генерації онтології повинен мати здатність доповнювати онтологію новими концептами та екземплярами, розбудовувати таксономічну структуру і нетаксономічні відношення між концептами, узгоджувати концепти, об'єднувати дві чи більше готових онтологій, поєднувати дані з різних джерел, відображати відмінності між концептами.

Критерій незалежності від сторонніх джерел інформації позначає потребу використання інструментом додаткових джерел інформації для генерації онтології. Такими джерелами можуть бути допоміжні онтології, словники, тезауруси, описи типів даних тощо. Бажаним є максимально можливе зменшення використання даних додаткових ресурсів.

Значна кількість засобів для побудови онтологій орієнтована на використання текстових джерел, описаних однією мовою (здебільшого англійською), особливо це стосується засобів, що використовують синтаксичний або лексичний аналіз. Таким чином, оптимальним рішенням був би інструмент, що потребував би мінімум адаптації при зміні мови тексту документів, на основі яких будується онтологія. Ми оцінюватимемо наявні методи відповідно до можливості реалізації такого рішення.

2.2. Вимоги до онтології

Згенерована онтологія має бути придатною до подальшого розвитку, зміни чи доповнення, зокрема за допомогою сторонніх інструментів.

Онтологія має відповідати загальноприйнятим стандартам, наприклад, зберігатися за допомогою поширених мов опису онтологій для уможливлення її повторного використання різними програмними засобами.

Мета побудови онтології – це не тільки забезпечення повного уявлення про структуру деякої предметної області, а й відображення подібності між концептами, сили зв'язків між ними, залежності між екземплярами та концептами, узгодження суперечностей, відображення семантики зв'язків в онтології, спільних властивостей концептів, а також можливу історію еволюції онтології.

3. Методи генерації онтологій

Для порівняння підходів і програмних засобів генерації онтологій пропонуємо спершу згрупувати ці інструменти за методами реалізації початкових етапів їх життєвого циклу:

Побудова онтологій шляхом перетворення XML-подібних документів на одну з мов опису онтологій. Такі методи дають можливість генерувати онтології на основі певного наявного коректного опису, здійсненого не тільки на мові XML, а й, наприклад, засобами MOF (Meta Object Facility) за допомогою стандарту XML Metadata Interchange. Це завдання здається, на перший погляд, тривіальним, проте проблемним місцем може бути потреба узгодження форматів, якщо онтологія будується шляхом злиття кількох XML-файлів (з подальшим перетворенням на одну з мов опису онтологій) від різних авторів. Метод найкраще надається до автоматизації через те, що маємо справу з надзвичайно структурованими джерелами даних.

Використання сторонніх ресурсів. Використання наявних лексичних баз даних (напр., Wordnet), словників (наприклад, Webster), тезаурусів (наприклад, НТОЕД) дає змогу не тільки здійснювати розширення і валідацію наявних онтологій, а також створювати онтології деякої предметної області «з порожнього аркуша». Однак найчастіше методи цієї категорії застосовують разом з іншими, заснованими на лінгвістичному аналізі текстів.

Застосування лінгвістичного аналізу текстів, написаних природною мовою. Одна з найпоширеніших груп методів генерації онтологій на основі аналізу набору неструктурованих текстових документів дає змогу генерувати як концепти онтології та їхні екземпляри й відношення. Можливе застосування ряду підходів, наприклад, використання лексико-синтаксичних шаблонів і граматики, граматичних особливостей мови, аналіз глибоких відмінків тощо. Однією з проблем цієї групи методів є їхня орієнтація на одну конкретну мову (здебільшого на англійську).

Застосування кластеризації та аналізу формальних концептів. Засоби агломеративної ієрархічної кластеризації текстових документів дають змогу виокремити групи, що відповідають концептам предметної області та побудувати таксономічні відношення між ними. Побудова відношення Галуа між частково-упорядкованою множиною об'єктів (наприклад, текстових документів) і частково-упорядкованою їхніх властивостей (наприклад, ключових слів із найбільшою частотою, видобутих із документів) також дає можливість отримати математичний об'єкт, придатний до серіалізації на одну з мов опису онтологій. Ми виділяємо ці дві групи методів в одну категорію, оскільки математичний апарат їхньої реалізації часто подібний. Попри ефективність цих методів на початкових етапах генерації онтологій, на етапах валідації та розширення виникає потреба залучення експерта або застосування інших методів (сторонніх ресурсів чи засобів лінгвістичного аналізу).

3.1. Перетворення XML-подібних документів

У праці [30] описано метод, що використовує два алгоритми, які виконують перетворення XML-документа згідно з моделлю сутність–відношення, а на основі утвореної моделі будує онтологію, використовуючи мову OWL. Ці перетворення дають змогу встановити на основі XML-описів класи онтології, атрибути класів, деякі види відношень та обмежень на значення атрибутів. Однак самі автори визнають потребу людського втручання в процес генерації онтологій на етапі визначення обмежень на атрибути даних. Валідність побудованої онтології визначається, зокрема, тим, чи правильно сформовано вхідний XML-файл. По суті, цей метод відповідає критерію автоматизованості частково на етапі аналізу та повністю на етапі серіалізації. Засобів розширення збудованої онтології цей інструмент не передбачає, проте побудована ним онтологія може бути розширена за використанням іншого інструменту.

В основі методу генерації онтології на основі XML Schema, описаного в [8], також лежить припущення про те, що деякий XML-документ містить реляційні структури, які можна перетворити на класи, екземпляри та властивості онтології. Відношення перетворюються в класи, стовпчики у властивості, а рядки – в екземпляри. Попри складність задачі виявлення неявних реляційних структур у XML-файлі, автори пропонують набір XSTL перетворень, що перетворюють деревовидну структуру XML-документа у трійку «об'єкт – предикат – значення» мови RDF. Якщо XML-документ не містить XML Schema, автори пропонують алгоритм утворення схеми

для такого документа. Варто зазначити, що тоді знижується рівень автоматизації процесу генерації через потребу внесення інформації про обмеження на значення даних. У випадку ж наявності XML Schema або використання тільки коректного файлу в цьому форматі генерація вихідної онтології відбувається автоматично. Засобів валідації та розширення онтології цей метод не передбачає. Такий метод використано, зокрема, при створенні платформи OntoWiki [4]. Схожий метод із детальнішим описом побудови нетаксономічних відношень в онтології, виходячи з XML Schema, описано в [6]. У [14] запропоновано метод перетворення XSD документів в онтології на мові OWL, однак із можливістю поєднання кількох вхідних XML Schema файлів, виявлення збігів між концептами і забезпечення коректного злиття онтологій ще на стадії аналізу.

Для ширшого практичного використання онтологій низка авторів пропонують застосування для побудови онтологій спеціалізованих засобів проектування програмного забезпечення, а саме UML (оскільки це найбільш поширена графічна мова для моделювання). Проте слід мати на увазі, що UML засновано на об'єктно-орієнтованій парадигмі і, відповідно, має певні обмеження стосовно побудови онтологій. Жодне з наявних рішень не задовольняє повне визначення онтології: визначення необмеженої за кількість рівнів ієрархії властивостей, моделювання екземплярів онтології, робота з множинними обмеженнями, визначення додаткових обмежень для властивостей (наприклад, властивість має певне значення) тощо. Відповідно, UML може використовуватися тільки на початкових фазах створення онтологій. Ці проблеми детально описано у [12; 15; 16]. Автори [12; 15] також пропонують використання XSLT для перетворення XMI (XML-подібної мови, на яку серіалізуються UML-діаграми чи інші засоби MDA) на OWL, а також оберненого перетворення, що дає можливість розширення чи зміни створеної онтології. Значний інтерес становлять праці [5; 13], автори яких описують подібності та відмінності між мовами UML і DAML (DARPA Agent Markup Language), а також можливість встановлення відповідності між концептами у цих мовах і проблеми, що виникають при цьому. Наведено також перелік бажаних розширень для UML для повної автоматизації прямого і зворотного перетворень.

Ця група методів є корисною для побудови онтології предметної області за наявності певної моделі спроектованої засобами MDA, однак очевидно, що в будь-якому випадку ці методи покладено на засоби, що не обходяться без людського втручання.

У [17] здійснено розробку засобу напіваавтоматичної генерації онтологій на основі видобування інформації з DTD-файлів. Це дослідження варто відзначити особливо, оскільки в ньому показано, що теги і структура XML-файлу дають достатньо інформації для побудови онтології. Цей метод однак потребує втручання експерта на стадії валідації хибно згенерованих зв'язків між концептами. Варто зазначити, що автори цього дослідження обмежуються тільки галуззю туризму, яка прогнозована з великою точністю, задля визначення релевантних концептів і щоб уникнути появи суперечностей. У дослідженні [19] описано процедуру автоматичного узгодження та встановлення відображення між двома DTD-файлами з подальшим утворення XML-файлу за їхньою схемою.

3.2. Використання готових словників

У [23] автори розробили метод на основі використання тезауруса WordNet. Автори розглядають WordNet як загальну онтологію, з котрої вони видобувають підмножину концептів для побудови онтології певної предметної області. Інструмент здійснює запит до WordNet щодо певного терміну і створює класи концептів за результатами запиту. Набір концептів, придатний до розширення, на кінцевій стадії серіалізується на мову OWL. Подібний метод описано у праці [22] з тією відмінністю, що автори цього дослідження використовують WordNet для розширення деякого набору пов'язаних термінів задля використання далі як допоміжної онтології.

Використання WordNet для валідації онтологій описано у студії [3]. Цей метод передбачає добування з тезаурусу групу термінів, пов'язаних із вхідним, з подальшим, запитом до Інтернет-документів щодо частоти зв'язків у цій групі термінів. Таким чином, терміни з низькою частотою можуть бути відкинуті, що призведе до поліпшення автоматично згенерованої онтології деякої предметної області. WordNet може бути також використаний для порівняння, узгодження та злиття двох онтологій, що описано в роботі [10]. У дослідженні [29] запропоновано використання EuroWordNet як засобу покращення генерованої онтології ще на стадії формування концептів. Проте слід зауважити, що недоліком такого підходу є те, що ми маємо справу радше з формою слова, а не його змістом. Цікава в цьому контексті праця [21], автори якої перетворюють словник Webster на графоподібну структуру, описану на мові XML, отримуючи деяку підмножину концептів, застосовують PageRank-подібний алгоритм для визначення залежності між концептами.

3.3. Використання засобів інтелектуального аналізу даних

Метод аналізу формальних концептів широко застосовують. Метод ґрунтується на теорії решіток, між якими встановлюється відповідність Галуа: описова решітка, що містить усі можливі описи (атрибути) екземплярів, упорядкованих щодо узагальнення, та решітки екземплярів, упорядкованих щодо включення, яке позначається. Дві відповідності між цими решітками (перша, що пов'язує кожен опис із множиною всіх об'єктів навчального набору, які потрапляють до нього, та відповідність, рівнозначна створенню найменш загального узагальнення опису всіх елементів деякої підмножини екземплярів) визначають відповідність Галуа між даними решітками. Пара аргумент-значення цієї відповідності становить концепт, множина якого впорядкована за узагальненням-специфікації.

Метод набуває дедалі більшої популярності для автоматичної генерації онтологій. Зокрема, у [24] аналіз формальних концептів використано для побудови онтологій на основі наборів текстових документів, він дає змогу згенерувати концепти, відношення належності екземплярів до концептів і таксономічні відношення. Різниця у підходах до застосування цього методу може полягати у різному виборі множини описів, якщо у [27] це слова, які найчастіше трапляються в навчальній вибірці, тоді у [20] множина ключових слів, отримана з документів, розширюється за допомогою тезауруса WordNet. Ця особливість визначає залежність методу від мови документу. Однак варто зауважити, результати такого процесу потребують валідації, оскільки кількість і зміст математично згенерованих концептів та відношень може відрізнятися від тієї, яку розпізнає людський розум, цей метод не придатний для модифікації онтології, натомість автори пропонують додавання нових даних до онтології, застосовуючи продукційні правила та наближений вивід.

Показує свою ефективність і метод ієрархічної кластеризації за наявності певної множини однорідних об'єктів із наперед відомим вектором ознак однакової довжини для всіх об'єктів. Показовим є приклад, описаний у [11]. Кластеризація здійснюється з використанням метрики категоризаційної корисності, розглянутої в [1], що визначає подібність об'єктів одного класу, і відмінність у різних. Автори подають також схему серіалізації утвореної ієрархії концептів на мову RDF. Попри те, що автори зазначають корисність цього методу в тих галузях людської діяльності, де бракує експертних знань. Можливості автоматизації розширення ієрархії концеп-

тів чи готової онтології даним методом, у згаданій праці не досліджено.

3.4. Застосування засобів аналізу природних мов

Ця група методів одна з найпоширеніших при автоматизації побудови онтологій.

Зокрема заслуговує на увагу праця [2], присвячена синтаксичному встановленню зв'язків, що передбачає виявлення термінів, пов'язаних із іншими термінами на основі визначених типів зв'язків у реченні або навіть цілих лексико-синтаксичних конфігурацій, визначених шаблонами, що задаються. Використання лексико-синтаксичних шаблонів для побудови і валідації онтологій пропонують автори [26], щоправда, варто зазначити, що побудова самих шаблонів потребує залучення експерта, що кардинально погіршує вимогу автоматизації. Загальною характеристикою цієї групи методів є те, що кожен із них призначений тільки для однієї конкретної природної мови (здебільшого англійської, хоча відомі розробки для російської, німецької або чеської).

Висновки

Розглядаючи можливість автоматизації різних стадій автоматичної генерації онтологій ми дійшли таких висновків.

Підготовчу стадію та стадію серіалізації у всіх випадках можливо повністю автоматизувати, оскільки ці процеси є повністю тривіальними і зводяться до примітивних операцій надрядковими даними або ж перетворення деревовидних структур даних у XML-подібний формат.

Стадію аналізу також можна ефективно автоматизувати. Побудова концептів, таксономічних відношень між концептами та відношень належності екземплярів до класів автоматизується під час застосування всіх описаних вище методів генерації онтологій. Можливість повної автоматизації побудови нетаксономічних відношень досі є відкритим питанням, більше того – наявні методи залежні від мови оброблюваних текстових даних. Відкрите також питання повної автоматизації найменування концептів, особливо це стосується методів автоматичної генерації онтологій, заснованих на використанні ієрархічної кластеризації та формального аналізу концептів. Проблему можна розв'язати завдяки використанню зовнішніх джерел знань, наприклад, тезаурусів чи словників на зразок WordNet.

Стадія валідації так чи так потребує втручання експерта. Виняток становлять випадки, коли онтологія генерується на основі набору XML-документів або ж підмножини записів якогось

словника чи тезауруса. У цьому контексті засоби на кшталт WordNet, без сумніву, заслуговують на особливу увагу через великі можливості для автоматизації валідації. Слід зауважити, що WordNet має занадто широке призначення і не може бути адаптованим до певної предметної галузі людської діяльності. Попри це, використання методу для валідації та узгодження онтологій є цікавою темою для подальшої розробки.

Стадія розширення також потребує особливої уваги в автоматизації. Особливо це стосується випадку її зведення до узгодження і злиття онтологій. Процес злиття онтологій тісно пов'язаний з узгодженням; нині існує низка методів його реалізації для двох вхідних онтологій, однак одночасне злиття кількох онтологій є відкритим питанням. Ця проблема може бути розв'язана послідовними злиттями, але в такому випадку кінцева онтологія буде залежати від вибору порядку злиттів. У деяких випадках додання нових сутностей і зв'язків в онтологію може відбуватися методом відмінним від того, який використовувався на початковому етапі (наприклад, аналіз формальних концептів на етапі

аналізу та використання продукційних систем на етапі розширення). Так чи інакше після стадії розширення онтології має здійснюватись повторна валідація, проблеми якої описані.

Отже, для досягнення максимальної ефективності обробки та систематизації інформації система керування знаннями, що використовує онтології має поєднувати різні методи автоматичної генерації. Генерація онтологій на основі XML-документів важлива через те, що такий формат став, по суті, стандартом для обміну даними між прикладними програмами. Великий інтерес становлять методи, засновані на аналізі формальних концептів та ієрархічній кластеризації з огляду на їхню більшу незалежність від мов текстових документів і потужну математичну основу. Для автоматизації стадії валідації ефективно використовувати сторонні словники і тезауруси, що призводить до потреби розробки україномовних відповідників WordNet. Ключовою складовою системи керування знаннями має бути засіб узгодження та злиття онтологій, що забезпечив би коректну інтеграцію знань.

Література

1. Ермаков А. Е. Автоматизация онтологического инжиниринга в системах извлечения знаний из текста / А. Е. Ермаков // Компьютерная лингвистика и интеллектуальные технологии : По материалам ежегодной Международной конференции «Диалог» (Бекасово, 4–8 июня 2008 г.). – Вып. 7 (14). – М. : РГГУ, 2008. – С. 154–158.
2. Рабчевский Е. А. Автоматическое построение онтологий на основе лексико-синтаксических шаблонов для информационного поиска / Е. А. Рабчевский. – Режим доступу: http://www.krc.karelia.ru/doc_download.php?id=2155&table_name=publ&table_ident=4499. – Назва з екрана.
3. Agirre E. Enriching very large ontologies using the WWW / E. Agirre, O. Ansa, E. Hovy, D. Martinez. – Режим доступу: http://ol2000.aifb.uni-karlsruhe.de/final/EAgirre_14.pdf. – Назва з екрана.
4. Auer S. OntoWiki – A Tool for Social, Semantic Collaboration / S. Auer, S. Dietzold, T. Riechert. – Режим доступу: http://www2007.org/workshops/paper_91.pdf. – Назва з екрана.
5. Baclawski K. Extending UML to Support Ontology Engineering for the Semantic Web / K. Baclawski, M. Kokar. – Режим доступу: <http://www.coe.neu.edu/~kokar/publications/baclawski.pdf>. – Назва з екрана.
6. Bedini I. Deriving Ontologies from XML Schema / I. Bedini, G. Gardarin, B. Nguyen. – Режим доступу: <http://arxiv.org/abs/1001.4901>. – Назва з екрана.
7. Bedini I. Automatic Ontology Generation : State of the Art / I. Bedini, B. Nguyen. – Режим доступу: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.62.5577>. – Назва з екрана.
8. Bohring H. Mapping XML to OWL Ontologies / H. Bohring, S. Auer. – Режим доступу: <http://www.informatik.uni-leipzig.de/~auer/publication/xml2owl.pdf>. – Назва з екрана.
9. Buitelaar P. Ontology Learning from Text Tutorial / P. Buitelaar, P. Cimiano, M. Grobelnik, M. Sintek. – Режим доступу: http://people.aifb.kit.edu/pci/OL_Tutorial_ECML_PKDD_05/ECML-OntologyLearningTutorial-20050923.pdf. – Назва з екрана.
10. Cho M. A new Method for Ontology Merging based on Concept using WordNet / M. Cho, H. Kim, P. Kim. – Режим доступу: <http://ieeexplore.ieee.org/iel5/10826/34122/01625891.pdf?arnumber=1625891>. – Назва з екрана.
11. Clerkin P. Ontology Discovery for the Semantic Web Using Hierarchical Clustering / P. Clerkin, P. Cunningham. – Режим доступу: <http://semwebmine2001.aifb.uni-karlsruhe.de/online/semwebmine03.pdf>. – Назва з екрана.
12. Djuric D. MDA-BASED ONTOLOGICAL ENGINEERING / D. Djuric, D. Gasevic, V. Damjanovic, V. Devedzic. – Режим доступу: <http://devedzic.fon.rs/publications/HANDBOOK2005.pdf>. – Назва з екрана.
13. Falkovych K. Stuckenschmidt H. UML for the Semantic Web : Transformation-Based Approaches / K. Falkovych, M. Sabou, H. Stuckenschmidt. – Режим доступу: http://www.cwi.nl/~media/publications/UML_for_SW.pdf. – Назва з екрана.
14. Ferdinand M. Lifting XML Schema to OWL / M. Ferdinand, C. Zirpins, D. Trastour. – Режим доступу: <http://vsiis-www.informatik.uni-hamburg.de/getDoc.php/publications/204/fzt-lxs-04.pdf>. – Назва з екрана.
15. Gasevic D. MDA-based automatic OWL ontology development / D. Gasevic, D. Djuric, V. Devedzic. – Режим доступу: <http://www.springerlink.com/index/8V6488UV524NG224.pdf>. – Назва з екрана.
16. Gasevic D. Ontology Modeling and MDA / D. Gasevic, D. Djuric, V. Devedzic. – Режим доступу: http://www.jot.fm/issues/issue_2005_01/article3.pdf. – Назва з екрана.
17. Giraldo G. Construction semi-automatique d'ontologies a partir de DTDs relatives a un meme domaine / G. Giraldo, C. Reynaud. – Режим доступу: <http://leo.saclay.inria.fr/publications/GEMO-PUBLI/display-abstract.php?id=244>. – Назва з екрана.
18. Gluck M. Information, uncertainty, and the utility of categories / M. Gluck, J. Corter. – Режим доступу: <http://www.citeulike.org/group/1788/article/1341486>. – Назва з екрана.
19. Hashmy R. Mapping Biological XML DTDs Using Ontology / R. Hashmy. – Режим доступу: <http://ww1.ucmss.com/books/LFS/CSREA2006/BIC3734.pdf>. – Назва з екрана.

20. Hotho A. Conceptual Clustering of Text Clusters / A. Hotho, G. Stumme. – Режим доступу: http://www.aifb.uni-karlsruhe.de/WBS/aho/pub/tc_fca_2002_submit.pdf. – Назва з екрана.
21. Jannink J. Thesaurus Entry Extraction from an On-line Dictionary / J. Jannink. – Режим доступу: <http://www-db.stanford.edu/pub/papers/f99.ps>. – Назва з екрана.
22. Kong H. Design of the automatic ontology building system about the specific domain knowledge / H. Kong, M. Hwang, P. Kim. – Режим доступу: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1625840. – Назва з екрана.
23. Moldovan D. Domain-Specific Knowledge Acquisition and Classification using WordNet / D. Moldovan, R. Girju. – Режим доступу: <http://www.seas.smu.edu/~rohana/papers/Flairs2000.ps>. – Назва з екрана.
24. Obitko M. Ontology Design with Formal Concept Analysis / M. Obitko, V. Snoel, J. Smid. – Режим доступу: <http://cite-seerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.2092>. – Назва з екрана.
25. Omelayenko V. Learning of Ontologies for the Web: the Analysis of Existent Approaches / V. Omelayenko. – Режим доступу: <http://www.cs.vu.nl/~borys/papers/WebDyn01.pdf>. – Назва з екрана.
26. Raghunathan, P. Fast semi-automatic generation of ontologies and their exploitation / P. Raghunathan. – Режим доступу: <http://people.cis.ksu.edu/~rpr/TR/1.pdf>. – Назва з екрана.
27. Thanh-Tho Q. Automatic Fuzzy Ontology Generation for Semantic Web / Q. Thanh-Tho. – Режим доступу: <http://doi.ieee-computersociety.org/10.1109/TKDE.2006.87>. – Назва з екрана.
28. Uren V. Semantic annotation for knowledge management: Requirements and a survey of the state of the art / V. Uren, P. Cimiano. – Режим доступу: <http://www.cs.vu.nl/~borys/papers/WebDyn01.pdf>. – Назва з екрана.
29. Wagner A. Enriching a Lexical Semantic Net with Selectional Preferences by Means of Statistical Corpus Analysis / A. Wagner. – Режим доступу: http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-31/AWagner_16.pdf. – Назва з екрана.
30. Xu J. Using Relational Database to Build OWL Ontology from XML Data Sources / J. Xu, Li W. – Режим доступу: <http://doi.ieee-computersociety.org/10.1109/CIS.Workshops.2007.139>. – Назва з екрана.

M. Glybovets, O. Bobko

THE METHODS OF AUTOMATIC ONTOLOGY GENERATION

In this study the classification of automatic ontology generation methods was performed and combination of different methods of automatic ontology generation for increasing efficiency of knowledge management system was analyzed.

Keywords: ontology, clustering, natural language processing, taxonomy, formal concept analysis, knowledge management.

Матеріал надійшов 03.04.2012