

Автоматизовані методи пошуку і відкриття необхідних сервісів

Розглянути переваги застосування технологій семантичного веб для опису предметної галузі та моделювання бізнес-процесів з допомогою онтології; для формування платформ виконання як композиції семантичних веб-сервісів, пошуку сервісів в репозитарії чи в базі знань за допомогою синтаксичного і семантичного порівняння описів відібраних сервісів (окремо чи в одній процедурі).

Ключові слова: системи сервісів, веб-сервіси, семантичні веб-сервіси, онтологія, профіль веб-сервісу, семантичний пошук сервісів

Вступ

Стрімке розповсюдження в всіх галузях сучасного суспільства індустрії сервісів, базованої на використанні сервіс-орієнтованої архітектури, передбачає створення найближчим часом розподілених сховищ з мільйонами сервісів, які можна було би знаходити і відкривати на основі вимог замовників прикладних додатків [2]. Найкращим варіантом для реалізації таких сервісів є веб-сервіси завдяки їх стандартизації і прив'язку до Інтернет.

Універсальна система URL-адресів і технологія гіпертексту в поєднанні з пошуковими програмами утворили середовище, де інформація не тільки передається, а й інтелектуально обробляється. Під інтелектуальною обробкою мається на увазі, зокрема, автоматизоване збирання (композиція) об'єктів (наприклад, сервісів) в множини за певними ознаками, встановлення відносин між цими множинами і т.д. Веб-сервіси представляються за допомогою спеціальних мов опису сервісів (наприклад, WSDL) у формі, що робить їх доступними і для людини, і для комп'ютера. Це дозволяє організувати

пошук і витяг веб-сервісів з універсального реєстру UDDI (Universal Description, Discovery and Integration) за їх синтаксичною інформацією при зіставленні ключових слів. Протокол SOAP використовується для того, щоб викликати сервіси і обмінюватися повідомленнями між додатками. Поєднання WSDL UDDI, SOAP полегшує застосування сервіс-орієнтованих концепцій в Інтернет, зокрема, для публікування, знаходження, пов'язування і виконання сервісів. Але все одно потрібна людина-програміст, який повинен розібратися в сервісах і зрозуміти їх зміст, або семантику, щоб вибрати потрібний сервіс з підмножини сервісів з подібними наборами ключових слів [3].

Семантичні веб-сервіси усувають цей недолік, і це великий крок вперед в напрямку спрощеного доступу до сервісів через мережу. Вони мають свої мови опису (наприклад, OWL-S) і повністю незалежні від платформи реалізації, оскільки базуються на використанні онтології - свого роду словника понять предметної області і сукупності явним чином визначених припущень щодо змісту цих понять. У найпростішому випадку онтології використовуються для підвищення точності пошуку сервісу і відкриття такого сервісу, в якому згадується в точності шукане поняття, а не довільних сервісів, в текстах опису яких зустрілося задане ключове слово [4]. Мета даної роботи полягає в розгляді застосування технологій семантичного веб для опису предметної галузі та моделювання бізнес-процесів, формування платформ виконання як композиції семантичних веб-сервісів, пошуку сервісів в репозитарії чи базі знань за допомогою

синтаксичного і семантичного порівняння описів відібраних сервісів (окремо чи в одній процедурі).

Веб-сервіси (WS) і семантичні веб-сервіси (SWS)

Пошук, або відкриття (discovery), сервісів є однією з найбільш невід'ємних властивостей сервіс-орієнтованих систем. Пошук полягає в ідентифікації сервісу з безлічі описів сервісів, що відповідають вимогам запиту на пошук. З збільшенням кількості сервісів, розроблених і пропонуваних в системі сервісів підприємства або в Інтернет, користувачі навряд чи в змозі вручну перевірити свої вимоги з знаходження необхідних сервісів. Користувачам можуть допомогти автоматизовані методи відкриття сервісів. Здатність ефективно знаходити сервіси залежить від того, як сервіси описані і як вони рекламуються, як можуть бути сформовані вимоги до сервісів і як ці вимоги перевіряються. Останнє спонукає до розроблення методів пошуку сервісів, які можуть бути застосовані в різних бізнес-процесах, забезпечуючи при цьому прийнятний компроміс між виразністю і ефективністю.

Пошук сервісів, які забезпечують бажану дію або реагування (за запитом користувача або за появою події) є одною з основних процедур в сервіс-орієнтованій архітектурі (SOA) і вважається однією з найважливіших задач у дослідженні веб-сервісів. Технології веб-сервісів в даний час є найбільш поширеним засобом реалізації сервіс-орієнтованих концепцій і призводять до становлення "Інтернет сервісів", де концентруються сервіси для всіх галузей життя і бізнесу. Сервіс є описом реалізації функціональності, що пропонується програмними компонентами. Оскільки сервіси слабко пов'язані і самодостатні, можна динамічно викликати і замінювати сервіси, наприклад, в бізнес-процесі, навіть через межі однієї компанії або організації. Зокрема, одним з конкретних застосувань є використання сервісів в

робочих потоках (workflows), що підтримують бізнес-процеси.

Щоб знайти веб-сервіс, необхідно спочатку описати елементи запропонованих і запитуваних сервісів та їх можливостей як можна більш точно. Для цього використовуються відомі технології веб-сервісів, такі як розширювана мова розмітки повідомлень XML (Extensible Markup Language), мова опису веб-сервісів WSDL (Web Service Description Language), протокол простого доступу до об'єктів SOAP (Simple Object Access Protocol). Пошук сервісів заснований на описі сервісних компонентів, наприклад, інтерфейсів або операцій, але успішність застосування та використання веб-технологій у пошуку сервісів залежить від здатності постачальників сервісів описувати сервіси, здатності користувачів формувати запити і від *порівнювача* описів і вимог (matchmaker), який реалізує алгоритми, що враховують запит і знаходять найбільш підходящі сервіси з набору сервісів компанії. Порівнювач фокусується на виборі необхідних елементів з опису сервісу, показниках подібності (similarity metrics) і комбінації отриманих значень подібності [6].

Часто виклику одного веб-сервісу не достатньо, щоб задовольнити особливі потреби або цілі замовників додатків. Наприклад, для покупки книги з Інтернет може знадобитися виклик веб-сервісу компанії кредитних карт і компанії доставки покупок додатково до сервісу самого книжкового магазину. Іншим прикладом є бронювання подорожей, де потрібно поєднувати сервіси кількох постачальників транспорту і сервіси готельного бізнесу. Необхідність виклику декількох сервісів для задоволення потреби вимагає певних координаційних заходів, які необхідно вирішувати. Основні підходи до координації веб-сервісів відомі як *оркестровка* та *хореографія*. При пошуку одного сервісу оркестровка пов'язана з внутрішньою імплементацією сервісу, а хореографія - з зовнішньою взаємодією між сервісом та

споживачем сервісу. Сьогоднішнім стандартом де-факто серед мов оркестровки є мова WS-BPEL (Web Services Business Process Execution Language), раніше відома як BPEL4WS [7].

Слід визнати, що веб-сервіси, описані з використанням мови WSDL, містять переважно *синтаксичну інформацію* і це затрудняє організацію автоматичного пошуку веб-сервісу та виконання композиції сервісів. Тому було запропоновано використовувати також *семантичну інформацію* у веб-сервісах, внаслідок чого з'явилося поняття семантичних веб-сервісів (SWS) з своєю мовою опису OWL-S (Web Ontology Language for Web Services). Методи, інструменти та технології Семантичного Веб (Semantic Web), який є продовженням сучасного Веб, зусиллями світового веб-консорціуму W3C (World Wide Web Consortium) спрямовані на погодження інформації в Інтернет з її семантикою і на забезпечення інтерпретації даних комп'ютерами. Кінцевою метою є автоматичне виконання завдання, надане комп'ютерові у вигляді адекватної інформації.

Семантичні веб-сервіси (SWS)

На початку XXI тисячоліття Тім Бернерс-Лі запропонував ідею Семантичного Веб [1], коли веб-сторінки оснащуються семантикою для того, щоб дати людині якісну змістовну інформацію і забезпечити шар даних, які комп'ютер може інтерпретувати. Зокрема Бернерс-Лі запропонував програмних агентів, які могли б працювати разом для того, щоб збирати і оброблювати (семантичні) дані. Він також визначив, що в «процесі, який називається відкриттям сервісу», треба було б описати семантичні функціональності агента з допомогою «спільної мови для опису сервісів таким чином, який дозволяє іншим агентам «розуміти» функції, що пропонуються, і шляхи, як скористатися цим».

SWS створюються на базі існуючої технології веб-сервісів для забезпечення динамічного пошуку сервісів, їх

композиції, виклику веб-сервісів, результатом чого є можливість автоматизації цих процедур, які в даний час вимагають втручання розробників програмного забезпечення. Підхід до рішення задачі полягає в використанні більш точних описів семантичних веб-сервісів, які зрозумілі для комп'ютера. Як правило, семантичний веб-концепт можна розділити на дані і метадані [3]. У той час, як дані відображають фактичний контент (зміст), метадані можуть бути визначені як "дані або інформація про дані". В цілому застосування метаданих має два аспекти: з одного боку, деталізується формат і організація представлення інформаційного змісту, а з іншого боку, - доменні знання пов'язуються з даними, які дозволяють будувати інтерфейси. Ключем до досягнення змісту, зрозумілому комп'ютерові, є групування термінів, що використовуються в описах метаданих, у чітко визначені, стандартизовані словники. Для цього, використовуються *онтології*, що мають стандартний формат.

Семантика в веб-сервісах використовується для придання деяким компонентам сервісів певного значення, наприклад, забезпечення детальної інформації про те, яку операцію, інтерфейс або повідомлення містить опис сервісу. Крім цього можливе визначення нових елементів, які не розглядалися в не-семантичних стандартах веб-сервісів. Передумови (preconditions) та наслідки (effects) є добре відомими прикладами таких нових компонентів.

Передумови і наслідки можуть бути логічними умовами, які повинні виконатися до того, як сервіс буде здійснено, і які описують відповідно зміни, які очікуються після виконання сервісу. На відміну від входів і виходів, які не мають станів, передумови та наслідки містять інформацію про стан до та після викликання сервісу. Разом з входами і виходами передумови та наслідки формують профіль сервісу [4].

Онтології і засоби їх побудови

В Семантичному Веб онтології забезпечують основу для формування комп'ютерно-зрозумілих даних і забезпечення обміну інформацією між людьми і комп'ютерами з врахуванням її синтаксичних і семантичних особливостей. Онтології можуть обслуговувати декілька доменів і бути описані в багатьох різних формах. Тому, як наслідок, немає ніякого загально прийнятого визначення терміну "онтологія", а наявні визначення відрізняються за науковими співтовариствами, які їх використовують. Наприклад, для комп'ютерних наук онтологія виконує функцію, подібну схемам баз даних, надаючи комп'ютерно-зрозумілу семантику інформаційним джерелам через зібрання термінів та співвідношень між ними. Семантика підтримує і забезпечує спільне розуміння знання домену, яке може бути передано між людьми та програмним забезпеченням.

В самому загальному випадку під онтологією розуміють формальний і явний опис спільної концептуалізації домену. Таке визначення підкреслює головні відмінні риси онтології: (1) необхідність певної мови для опису онтології; (2) використання абстракцій галузі, які включають в себе відповідні поняття; (3) узгоджену точку зору на представлення основ знань домену. Тобто онтологію слід трактувати як структурну специфікацію деякої предметної галузі, її формалізоване уявлення, яке включає словник (або імена) покажчиків на терміни цієї галузі і логічні вирази, які описують, як вони співвідносяться один з одним. Такий словник підтримує обмін знаннями про деяку предметну галузь і безліч зв'язків, встановлених між термінами в цьому словнику [9].

Відносини між семантичними поняттями зазвичай визначаються за правилами логічного виводу. Консорціумом W3C (World Wide Web Consortium) розроблені два основних стандарти для представлення онтологій, а саме: RDF Schema (RDFS) і мова OWL (Web Ontology Language). RDFS і OWL

засновані на RDF (Resource Description Framework), стандарту на основі XML, який дозволяє визначити трійки (триплети). Трійки визначають відношення між довільними суб'єктом і об'єктом за допомогою так званого предикату. В RDF і OWL суб'єкти, об'єкти і предикати визначаються через однакові ресурсні ідентифікатори URI (Uniform Resource Identifier), або відповідно багатомовні ресурси ідентифікаторів (Iris). Таким чином, стандарт RDF забезпечує дуже універсальний і гнучкий механізм для представлення знань [3-5]. Семантичний Веб існує, базуючись на принципі мінімальних обмежень: чим менш правил, тим краще. Це означає, що Семантичний Веб надає велику свободу вислову і тому будь-хто може говорити про будь-що. Коли ретельно дослідити те, що робить Семантичний Веб, стає очевидним, чому саме такий рівень можливостей потрібний. Якби почали обмежувати людей, вони були б не в змозі будувати бажаний ряд додатків, і Семантичний Веб виявилася б для багатьох даремним. Оскільки RDF використовує URI, інформація повністю децентралізована, не потрібно звертатися за дозволом до деякого центрального органу, щоб викласти свою інформацію. Останніми роками розробка онтологій - формальних явних описів термінів предметної галузі і відносин між ними - перейшла зі світу лабораторій з штучного інтелекту на робочі місця експертів з предметних галузей. В багатьох дисциплінах зараз розробляються стандартні онтології, які можуть використовуватися експертами з предметних галузей для сумісного використання і анотування інформації в своїй галузі. Сумісне використання людьми або програмами загального розуміння структури інформації є однією з найбільш загальних цілей розробки онтологій [9]. Наприклад, нехай декілька різних веб-сайтів містять інформацію з медицини або надають інформацію про платні медичні послуги, оплачувані через Інтернет. Якщо ці веб-сайти спільно вживають і публікують одну і ту ж базову

онтологію термінів, якими вони всі користуються, то комп'ютерні агенти можуть витягувати інформацію з цих різних сайтів і накопичувати її для відповідей на запити користувачів або як вхідні дані для інших застосувань.

Онтологія - це формальний явний опис понять в даній предметній галузі: **класів** (іноді їх називають поняттями), **властивостей** кожного поняття як його атрибутів (іноді їх називають слотами), і **обмежень**, накладених на слоти (іноді їх називають фацетами). Онтологія разом з набором індивідуальних екземплярів класів утворює *базу знань*. Класи описують поняття предметної галузі. Наприклад, клас Action може представляти всі процедури (запуску завдань, передачі даних, контролю потоку даних та інше). Конкретні процедури – це **екземпляри** цього класу. Процедура передачі даних чи запуску завдання - це екземпляр класу Action. Клас може мати підкласи, які представляють більш конкретніші поняття, ніж надклас. Наприклад, ми можемо розділити клас всіх Action на Action_Group (послідовність процедур) і на Task (одне завдання).

Атрибути описують властивості класів і екземплярів: процедура Task може містити файл (*containsFile*), створювати ресурс (*createsResource*) чи залежити від певних умов (*hasDependency*). Атрибути можуть мати різні фацети, які описують тип значення, дозволені значення, число значень (потужність) і інші властивості значень, які може приймати слот.

При цьому слід дотримуватися наступних фундаментальних правил розробки онтології:

- Не існує єдиного правильного способу моделювання предметної галузі - завжди існують життєздатні альтернативи. Краще рішення майже завжди залежить від передбачуваного застосування і очікуваних розширень.
- Розробка онтології - це обов'язково ітеративний процес.

Поняття в онтології повинні бути близькі до об'єктів (фізичних або логічних) і

відносин в тій предметній галузі, для якої створюється онтологія. Швидше за все, це іменники (об'єкти) або дієслова (відносини) в пропозиціях, які описують вибрану предметну галузь.

Із поширенням семантичних технологій та їх упровадженням, зростає і кількість інструментів для менеджменту їх центрального елементу – онтологій. Загалом, існує біля 100 розробок, серед яких найпопулярнішими є Protégé 2000/Jena, PCPACK, Ontolingua, OntoStudio, Ontosaurus/PowerLoom, Apollo CH, CEO [8,10]. Для прикладу на рис.1 наведено головне вікно редактора Protégé, в якому виділені базові процедури побудови онтології.

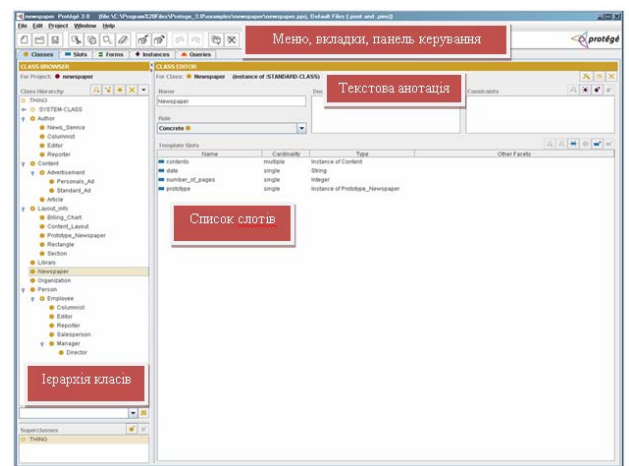


Рис.1. Редактор Protégé

Онтології і засоби їх опису

Перше покоління засобів відкриття сервісів було засновано на централізованих методах виявлення (наприклад, в реєстрі UDDI), коли веб-сервіси описані через функції своїх інтерфейсів. Реєстр UDDI не відображає відносини між суб'єктами в своєму каталозі і тому не спроможний використати семантичну інформацію для пошуку цих взаємних відносин. Механізм пошуку UDDI підтримує лише прямі порівняння описів. У тих випадках, коли немає прямих посилань і необхідно скласти набір сервісів для виконання запиту, UDDI не забезпечує ніяких результатів пошуку, тому що він не може вийти за рамки прямих порівнянь описів сервісів на базі порівняння

ключових слів. Мові WSDL не вистачає семантики, яка необхідна для того, щоб комп'ютери зрозуміли, яке призначення сервісу. Тому робилися і продовжуються спроби удосконалення існуючої технології WSDL і UDDI, надавши веб-сервісам комплексні семантичні анотації [7,20]. Але відносини між реальними поняттями (концептами) онтологій настільки складні, що краще на пряму використовувати мову OWL-S, як мову представлення онтологій для опису семантичних відносин між поняттями, і організувати пошук веб-сервісів в розподіленому репозитарію.

Основу онтології веб-сервісу складають три взаємопов'язані під-онтології, що мають назви «профіль» (profile), «модель процесу» (process model) та «прив'язка» (grounding), рис.2. Онтологія профілю використовується для вираження того, «що робить сервіс» для потреб рекламування, побудови запитів, пошуку відповідників на запит. Модель процесу описує, «як сервіс працює» для уможливлення активації, взаємодії, компонування, моніторингу, відновлення. Прив'язка відображує компоненти моделі процесу з врахуванням детальної специфікації форматів повідомлень, протоколів і т.д. Усі ці під-онтології поєднані у концепт верхнього рівня «Сервіс» (Service), що посилається на вищезгадані три складові (ServiceProfile, ServiceModel, ServiceGrounding) відповідно через властивості типу «представляє» (presents), «описаний» (describedBy), «підтримує» (supports), рис.2.

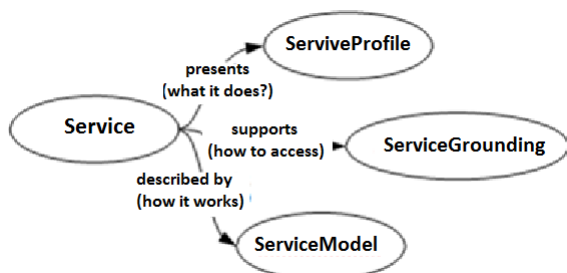


Рис.2. Шаблон онтології семантичного веб-сервісу

Найбільш важливим типом інформації, яку представляє профіль і яка відіграє ключову роль у процесі виявлення сервісу, є аспекти функціональності сервісу: перетворення інформації, представлене *входами* та *виходами* сервісу, та зміна станів сервісу під час виконання, представлена *передумовами* та очікуваним *ефектом*. Таким чином функціональні властивості описують входи, виходи, передумовами і ефекти сервісу, тобто IOPE – inputs / outputs / preconditions / effect. Нефункціональні властивості описують напівструктуровану інформацію, призначену для людини-користувача, наприклад, ім'я сервісу і параметри сервісу, що включають вимоги до безпеки і якості сервісу, географічні дані і інше.

Семантична розмітка сервісів дозволяє організувати наступне: *відкриття* сервісів (discovery), їх *вибір* (selection), *композицію* (composition), *здійняти у посередництві* (mediation), *здійснити їх виконання* (execution), *моніторинг* (monitoring), *заміщення* (replacement) і *компенсацію* (compensation), щоб забезпечити налаштування деякої задачі. Різні кроки, необхідні для досягнення такого налаштованого завдання, називають *виконавчим процесом семантичних веб-сервісів*. Базовою процедурою є відкриття сервісу, що відповідає процесу знаходження відповідного сервісу в наборі сервісів, які можуть бути надані центральним або розподіленим репозитарієм сервісів. Після виявлення кількох відповідних сервісів можна вибрати найбільш підходящий сервіс і запустити / послатися на нього. Першим кроком у процесі відкриття сервісу є порівняння категорій сервісів. Категорія сервісу зберігається в ServiceProfile (рис.3). Це значення порівнюється з категорією сервісу з користувацького запиту. Якщо виявлена відповідність категорій, то веб-сервіс вибирають для участі в наступному етапі процедури відкриття.

Ступень відповідності функціональності сервісів підраховується на другому етапі визначення відповідності функціональ-

ності сервісів. Атрибути - *hasInput*, *hasOutput*, *hasPrecondition* і *hasResult* зіставляються з тими, що містяться в запрошеному користувачем сервісі. Ці атрибути визначені в *ServiceProfile*. На третьому етапі обчислюється ступень відповідності якостей сервісів (QoS), яка визначається на основі оцінювання часу відгуку і надійності системи виявлення сервісів. Після виконання трьох зазначених етапів публікується лист сервісів, які найкраще підходять для задоволення запиту [12].

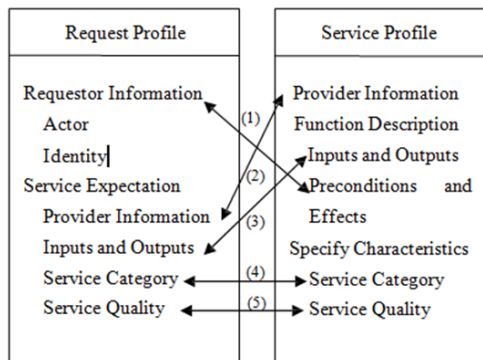


Рис.3. Процес виявлення сервісу за запитом

Семантичні підходи дозволяють виявити схожість (близькість) описів, навіть якщо в них є синтаксичні відмінності. Можливості сервісу описані за допомогою класу *ServiceProfile*, тому наявний і замовлений сервіси порівнюються на основі їх профілів. Алгоритм порівняння збігається, коли всі виходи замовленого сервісу збігаються з виходами наявного, і всі входи наявного збігаються з входами запиту. Це гарантує, що наявний сервіс задовольняє всі вимоги запиту. Існуючі підходи семантичного пошуку на основі онтології, як правило, базуються на оцінках концептуальної близькості між елементами онтології. Якщо сервіси і запиту представлені як набори триплетів, то виконується обчислення семантичної близькості між триплетами. Оцінкою близькості між сервісом і запитом є

числове значення, яке виражає ступінь подібності між ними.

Ведемо наступні спрощені позначення компонентів онтології: онтологія *O* являє собою кортеж

$$O = \langle C, E, R, T \rangle,$$

де *C* - безліч понять (класів); *E* - безліч екземплярів понять; *R* - безліч предикатів - типів відносин; *T* - безліч відносин, які включають різні види зв'язку між класами, екземплярами, предикатами, включаючи частковий порядок на множинах *C* і *R*, що задає відносини типу *is-a* - «підклас-суперклас» - «вище-нижче».

Облік структури онтології і семантики відносин дозволяє обчислювати оцінки семантичної близькості між елементами онтології (поняттями, екземплярами, предикатами). Ці оцінки близькості називаються елементарними оцінками близькості, на основі яких визначаються близькості між триплетами. Оцінки близькості між триплетами потім використовуються для визначення близькості між метаописами сервісів: 1) сума близькості між складовими триплетами; 2) максимальне паросполучення в підвищеному дводольному RDF графі онтології [17]. Сервіс *d* вважається релевантним заданому запиту *q*, якщо і тільки якщо оцінка семантичної близькості між ними перевищує деяку порогову величину. На сьогодні розроблено багато алгоритмів для елементарних оцінок близькості між компонентами триплетів, між триплетами і між метаописаннями [18].

Нижче для прикладу приведений дуже спрощений алгоритм оцінки відповідності сервісів за порівнянням вихідних параметрів. Якщо *OutR* представляє вміст (концепт) виходу замовленого сервісу, і *OutA* - виходу наявного сервісу з репозитарію, то можливі чотири ступені збігу, визнані алгоритмом:

- *Точний збіг*: якщо *OutR* і *OutA* такі ж самі або *OutR* є безпосереднім підкласом *OutA*. Це найкращий рівень порівняння.

- *Підключення*: якщо *OutA* включає *OutR*, то *OutA* може бути підключений замість *OutR*. Це наступний кращий рівень

порівняння, оскільки повернутий виходу може бути використаний замість того, що очікується від замовленого сервісу.

- *Неясність*: якщо *OutR* включає *OutA*, але постачальник може або не може повністю задовольнити відповідним вимогам запиту.

- *Невдача*: якщо немає співвідношення між *OutA* і *OutR*.

Найгірший рівень збігу між виходами / входами обирається як глобальний рівень порівняння між виходами / входами замовленого і наявного сервісів. Наявні в репозитарії сервіси упорядковуються за найбільшою кількістю балів у виходах. Входи порівнюються тільки тоді, коли існують однакові бали для виходів. Порівнювач онтологій попередньо обчислює співвідношення між концептами, що представляють входи і виходи всіх наявних сервісів в базі онтологій при їх реєстрації. Коли приходить запит, відповідний механізм (двигун) просто перевіряє зв'язки між концептами виходів / входів наявних сервісів і запитаного сервісу.

Окрім семантичного методу відкриття веб-сервісів, розглянутого вище, існують ще декілька інших методів: контекстно-базоване виявлення, кластерне виявлення, дедуктивне виявлення [13-17]. Після розгляду наявних методів оцінки семантичної подібності сервісів було запропоновано для реалізації в експериментальному прототипі сервісної системи інжинірингу застосувати гібридний підхід, який передбачає пошук веб-сервісів в два етапи:

- на першому етапі за допомогою синтаксичного порівняння сервісів звужується область пошуку шляхом відсіювання непричетних до запиту сервісів або виділенням одного з кластерів, сформованих застосуванням кластерного методу;
- на другому етапі проводиться уточнення попереднього вибору застосуванням семантичного порівняння описів відібраних сервісів із запитом, для чого складаються семантичні анотації для

веб-сервісів або описи онтологій для семантичних веб-сервісів.

Висновки

Загальновідомо, що в різних предметних областях одні й ті ж поняття можуть бути представлені різними термінами. Механізм онтології в цих випадках дозволяє формувати осмислені ієрархічні взаємозв'язки між сервісами, тобто реалізувати композицію сервісів, здатну задовольнити запит користувача, хоча в її описі може не бути деяких ключових слова з вхідного запиту. Методи пошуку семантичних веб-сервісів засновані на оцінках концептуальної близькості між елементами онтології сервісів (розміщених в реєстрі) і пошуковому запиту, заданому у вигляді множини триплетів. Оцінки близькості між триплетами потім використовуються для визначення близькості між метаописами сервісів. Механізми пошуку можуть застосовувати онтології для вибірки сервісів з синтаксично різними, але семантично однаковими словами. Онтології можуть використовуватися для організації обміну даними та інтеграції програм, а також побудови бази знань предметної галузі (наприклад, електротехніки).

Література

1. Tim Berners-Lee, James Hendler, and Ora Lassila. *The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities* // *Scientific American*, pp. 34–43, May 2001.
2. Петренко А.А. Цели и объекты науки о сервисах // *Системный анализ и информационные технологии*, №1, 2015.- С. .
3. Петренко А.І., Булах Б.В., Хондюл В. В. *Основи семантичного Веб / Вид-во «Політехніка», 2012.- 169 С.*
4. Cary Pennington, Jorge Cardoso, John A. Miller, Richard Scott Patterson, and Ivan Vasquez. *Introduction to Web Services. In Jorge Cardoso, editor, Semantic Web Services: Theory, Tools, and Applications, Premier Reference Source, chapter VII, pages 134–154. IGI Global, Hershey, PA, USA, 2007.*
5. Seppo Torma , Jukka Villstedt, Ville Lehtinen, Ian Oliver, Vesa Luukkala. *Semantic web services — a*

- survey, Helsinki University of Technology, 2008. - <http://www.cs.hut.fi/Publications/Reports/B158.pdf>
6. Katia P. Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan. Automated discovery, interaction and composition of Semantic Web services // *Journal of Web Semantics*, 1(1): 27–46, 2003
 7. Jacek Kopecký, Tomas Vitvar, Carine Bournez, and Joel Farrell. SAWSDL: Semantic Annotations for WSDL and XML Schema // *IEEE Internet Computing*, 11(6):60–67, 2007.
 8. Protege (2000). The Protege Project- <http://protege.stanford.edu>
 9. Natalya F. Noy & Deborah L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology* // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
 10. Duineveld, A.J., Stoter, R., Weiden, M.R., Kenepa, B. and Benjamins, V.R. (2000). A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies*, 52(6):1111-1133.
 11. Soodeh Pakari, Esmaeel Kheirkhah and Mehrdad Jalali. Web service discovery methods and techniques: a review // *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*, Vol. 4, No.1, February 2014, 14 p.
 12. Hui Peng, Zhongzhi Shi, Lirong Qiu. Matching Request Profile and Service Profile for Semantic Web Service Discovery.- <https://www.aaai.org/Papers/Workshops/2007/WS-07-11/WS07-11-010.pdf>
 13. Banage T. G. S. Kumara, Incheon Paik, Koswatte R. C. Koswatte, Wuhui Chen. Improving web service clustering through post filtering to bootstrap the service discovery // *International Journal of Services Computing (ISSN 2330-4472)*, Vol. 2, No. 3, July – Sept. 2014.
 14. Yu Yue Du, Yong Jun Zhang and Xing Lin Zhang. A Semantic Approach of Service Clustering and Web Service Discovery // *Information Technology Journal* 12 (5): 967-974, 2013 ISSN 1812-5638, I DOI [10.3923/itj.2013.967.974](https://doi.org/10.3923/itj.2013.967.974)
 15. Huiying Goa, Wolffried Stucky and Lei Liu. Web Services Classification Based on Intelligent Clustering Techniques // *International Forum on Information Technology and Applications*, Vol. 3, pp. 242-245, 2009.
 16. Richi Nayak and Bryan Lee. Web Service Discovery with additional Semantics and Clustering // *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 555-558, 2007.
 17. Крюков К.В., Панкова Л.А., Пронина В.А., Шипилина Л.Б. Меры семантической близости в онтологиях.- // *Проблемы управления*. – 2010. – №2. – С. 2–14.
 18. Нгуен Ба Нгок, Тузовский А. Ф. Модель информационного поиска на основе семантических метаописаний. -// *Управление большими системами*. Выпуск 41.-2014.- С.51-92.
 19. Муромцев Д.И. *Онтологический инжиниринг знаний в системе Protégé* // СПб.: СПб ГУ ИТМО, 2007.
 20. Антонов И.В., Воронов М.В. Формирование онтологических моделей предметной области для электронных обучающих систем // *Информационные технологии в обеспечении нового качества высшего образования. Сборник научных статей*. – Кн. 2. – М.: Исследовательский центр проблем качества подготовки специалистов, 2010. – С. 48–55.115.