

# ЗАСТОСУВАННЯ ДЕСКРИПТИВНИХ ЛОГІК ІЗ КОНКРЕТНИМИ ДОМЕНАМИ ДО ВИРІШЕННЯ ЗАДАЧ ВИЯВЛЕННЯ ТА КОМПОЗИЦІЇ СЕМАНТИЧНИХ ВЕБ СЕРВІСІВ

Ольга Захарова

Дане дослідження є частиною загальної науково-прикладної задачі, а саме, проблеми використання та розробки онтологічних підходів із застосуванням апарату дескриптивних логік (ДЛ) для ефективного автоматизованого вирішення складних бізнес-задач, що базуються на сервіс-орієнтованій архітектурі. В основу запропонованого підходу покладено ідею максимальної онтологізації, тобто всі аспекти проблеми визначаються засобами ДЛ онтологій. Запропонована онтологічна система спирається на п'ять базових типів онтологій, а саме: (1) онтологію загальних прикладних понять для визначення характеристик контексту, (2) онтологію прикладного домену для уніфікації та семантизації області застосування веб-сервісів, (3) загальну онтологію веб-сервісу верхнього рівня для ДЛ формалізації визначення сервісу як об'єкта в цілому, (4) онтології задач веб-сервісів, які забезпечують формальне представлення саме відповідних задач та (5) онтології конкретних сервісів, що забезпечують уніфікований формально визначений семантизований ресурс, який у сукупності з вище названими онтологіями уможливує автоматичне вирішення задач веб-сервісів. Фокусом цього дослідження є проблема застосування дескриптивних логік із конкретними доменами до формалізації описів таких базових задач семантичних веб-сервісів, як їх виявлення та композиція. В роботі визначені онтології задач виявлення та композиції семантичних веб-сервісів як розширення базової онтології веб-сервіса верхнього рівня спеціальними складними концептами, елементами яких є пари сервіс-запит/сервіс-сервіс, які співставляються відповідно до задачі, що вирішується. А також на їхній основі, визначені спеціальні концепти, що уособлюють множини встановлених подібностей між наявними сервісами та пошуковим запитом у задачі виявлення, або відповідностей сервісів типу «попередник-послідовник» для побудови ланцюжка композитного сервісу. Запропоновані твердження TBox побудовані за допомогою дескриптивної логіки з конкретними доменами  $\mathcal{ALL}(\mathcal{D})$ . Для цього визначений конкретний домен як пара: (1) об'єднання сукупностей можливих підмножин множини індивідів концепту *Parameter* й можливих підмножин множини індивідів концепту *Context*, та (2) визначений на ньому набір предикатів. Наведені в роботі онтології задач веб-сервісів охоплюють спрощену модель представлення веб-сервісів та передбачають подальше вдосконалення шляхом розширення моделі представлення сервісу, що розглядається, та залучення додаткових характеристик як функціональної так і процесної моделей представлення до вказаних онтологій задач. Це безперечно підвищить точність отриманого результату, але є методом подальших досліджень.

**Ключові слова:** задачі семантичних веб-сервісів, виявлення веб-сервісів, композиція веб-сервісів, цільовий запит, дескриптивні логіки з конкретними доменами, конкретний домен, предикатні символи, відповідність веб-сервісів, ланцюг сервісів типу «попередник-послідовник», функціональна модель веб-сервісу, процесна модель веб-сервісу, роз'язуваність конкретного домену, контекстний опис сервісу, відповідність веб-сервісів за контекстами, онтологізація вирішення проблеми, система онтологій, загальна онтологія веб-сервісу верхнього рівня, онтологія домену, онтологія задачі веб-сервісу, таксономія іменованих сутностей, онтологічна структура загальних понять.

This research is one of the branches of a general scientific and applied problem, namely, the problem of using and developing ontological approaches with descriptive logics (DL) apparatus to obtain the effective automated solution of complex business task based on service-oriented architecture. The base idea of the proposed method is the maximal ontologization, that is, all aspects of the task are determined by means of DL ontologies. The proposed ontology system is based on five basic types of ontologies, namely: (1) an ontology of general entities for defining the characteristics of the service's context, (2) an ontology of the application domain to formalize and unify semantical descriptions of web services, (3) the top-level ontology of web services for the DL formalization of the definition of the web services in general, (4) the ontologies of the web services tasks, which provide a formal representation of the tasks on all stages of web services life cycle, and (5) the ontologies of specific web services. All listed ontologies are related. Such, ontologies of the last type include links to all other above-mentioned ontologies and, together with them, provide a unified formally defined semantic resource, which enables the automatic solution of web service tasks. The focus of this study is the applying descriptive logics with concrete domains to the formalization of such basic semantic web services tasks as the web services discovery and composition. In this article, the ontologies of the web services tasks are defined as an extension of the general top-level web service ontology with special complex concepts. Their elements are pairs of "web service-goal (request)"/"web service-web service", which are compared according to the task being solved. Also, on their basis, special complex concepts are determined, which specify a set of found similarities between existing services and a goal in the discovery task, or «predecessor-follower» type responsibilities of web services to build a composite service. The proposed TBox statements are constructed using descriptive logic with concrete domains  $\mathcal{ALL}(\mathcal{D})$ . For this purpose, a concrete domain is defined as a pair of sets: (1) the union of sets of possible subsets of the set of individuals of the *Parameter* concept and possible subsets of the set of individuals of the *Context* concept, and (2) a set of predicates defined on it. Such definition of the concrete domain provides possibility of using only functional roles to construct the DL statements of complex concepts *DiscoveryServices* and *CompositionObject* for determining corresponding web services tasks. Note, the defined tasks and web services ontologies cover only simplified model of the representation of web services, in particular, functional *Input/Output* model extended by contexts descriptions. They provide the basis for further improvement by expanding this model and involving additional characteristics of both functional and process models of representation to the specified task ontologies. This will certainly increase the accuracy of the obtained result, but is the subject of further research.

**Keywords:** semantic web services problems, a discovery task, a composition task, a discovery goal, composition goal, description logics with concrete domains, concrete domain, predicate symbols, web services matching, a «predecessor-follower» chain, a functional model of web services, a process model of web services, concrete domain solvability, a service context, web services contexts matching, ontological approaches for solving problems, system of the ontologies, top level general web service ontology, a domain ontology, an ontology of the web services task, named entities taxonomy, ontological structure of general entities.

## Вступ

Онтології дедалі більше охоплюють усі сфери, пов'язані не лише з представленням даних та знань, а й з визначенням об'єктів, процесів, станів, задач та навіть алгоритмів. Це обумовлено їхньою ефективністю як інструмента формального опису елементів інформації та наявністю механізмів міркування, що дозволяють виводити нові знання та робити певні висновки з наявних тверджень.

На сьогодні, онтології є найпоширенішим інструментом для формального опису прикладних областей або доменів. Що ж до семантичних веб-сервісів, відомо чимало досліджень, щодо застосування онтологічних формалізмів, зокрема, апарату дескриптивних логік, для вирішення задач представлення та семантизації сервісів. Таке формальне представлення дозволяє не лише запобігти неоднозначності у розумінні семантики під час аналізу наявних веб-сервісів, а надає можливості автоматизованого вирішення їх задач, а саме: пошуку, виявлення, верифікації, композиції, тощо. Використання інтегрованої онтології домену дозволяє перейти до однозначно розуміти семантики значущих елементів прикладної області. Це забезпечує достовірність отриманих результатів пошуку веб-сервісів. А специфікація загальної онтології семантичного веб-сервісу, як об'єкта досліджень, надає єдиний формальний опис головних характеристик сервісу, які є значимими для вирішення задач.

## Задачі виявлення та композиції веб-сервісів

Не формально, проблема виявлення є задачею пошуку веб-сервісів, що відповідають запиту з максимальним ступенем подібності, де запитом, фактично, є гіпотетичний віртуальний сервіс, який втілює вимоги запитувача. Водночас подібність повинна розглядатися та оцінюватися як складне багатомірне поняття, тому що дуже рідко можливо досягти стовідсоткового збігу. Адекватність отриманого результату значною мірою залежить від точності та повноти формального представлення як мети пошуку (пошукового запиту), так і наявних сервісів. Семантичні веб-сервіси визначаються на двох рівнях: функціональному – своїми основними характеристиками, та процесному, що містить порядок, умови та результати виконання операцій всередині сервісу.

Залежно від множини характеристик сервісів, що визначаються, існує кілька різновидів функціональної моделі. Найпоширенішою з них є IOPE модель, яка описує сервіс чотирма наборами: вхідних (Input), вихідних (Output) параметрів, передумов (Precondition) та ефектів (Effect) сервісу. Наочно, IO (Inputs-Outputs) надають синтаксичну інформацію про веб-сервіси, тоді як PE (Preconditions-Effects) – відображають їх семантику. Саме цю модель представлення обрано базовою для подальшого вирішення задач.

Враховуючи той факт, що у специфікації сервісу закладено можливість визначення його стислого текстового опису (тег Documentation у WSDL-описі), який, фактично, може містити відомості про його призначення та функціональність (контекст сервісу), слід зазначити, що ця інформація може відігравати не останню роль у процесі співставлення сервісу з метою. Принаймні, змістовний структурований контекст може дозволити на початковому етапі вирішення задачі суттєво скоротити множину сервісів, що будуть підлягати подальшій обробці, через виключення з неї сервісів зовсім іншого призначення та функціональності.

Так, розширивши IOPE модель [1,2] контекстним описом [3], можемо визначити сервіс як:

$$S = \{In_s, Out_s, Pre_s, Eff_s, Doc_s\}, \text{ де} \quad (1)$$

$In_s$  – кінцева множина вхідних параметрів сервісу,

$Out_s$  – кінцева множина вихідних параметрів сервісу,

$Pre_s$  – кінцева множина передумов сервісу,

$Eff_s$  – кінцева множина ефектів сервісу,

$Doc_s$  – кінцева множина значимих семантичних елементів контекстного опису сервісу.

Мета виявлення, або пошуковий запит є описом віртуального сервісу, що уособлює вимоги запитувача, та може бути представлений як:

$$Q = \{In_q, Out_q, Pre_q, Eff_q, Doc_q\}, \text{ де} \quad (2)$$

$In_q$  – кінцева множина вхідних параметрів шуканого сервісу,

$Out_q$  – кінцева множина вихідних параметрів шуканого сервісу,

$Pre_q$  – кінцева множина передумов шуканого сервісу,

$Eff_q$  – кінцева множина ефектів шуканого сервісу,

$Doc_q$  – кінцева множина значимих семантик опису шуканого сервісу.

Відповідно до наведеної моделі, задачу виявлення Web-сервісу на функціональному рівні можна визначити як семантичний пошук та виявлення веб сервісів на основі покрокового порівняння описів наявних в реєстрі сервісів з цільовим запитом, що є описом абстрактного шуканого сервісу. Метою даного етапу є знаходження веб сервісу, що реалізує поставлену бізнес-задачу, або формування множини сервісів кандидатів для подальшої побудови композитного сервісу, а саме:

- визначення цілі виявлення як пошукового запиту, що є представленням віртуального бажаного сервісу;
- аналіз контексту та категоризація запиту та оголошених сервісів;
- співставлення (matching) контекстних описів пошукового запиту та сервісів з метою скорочення множини сервісів, що аналізуються;
- синтаксичне співставлення сервісів й цілі за вхідними та вихідними параметрами;
- перевірка відповідності передумов та ефектів цілі виявлення (goal discovery), перевірка істинності передумов;

- f. верифікація веб сервісу – перевірка відповідності сервіса його специфікації за допомогою резонерів ДЛ;
- g. перевірка нефункціональних характеристик сервісу [4], таких як вартість, повноваження, ступінь довіри до сервісу, репутація сервісу та інші. Слід зазначити, що врахування нефункціональних характеристик сервісу підвищує якість відбору сервісів, але суттєво ускладнює задачу;
- h. встановлення ступеня семантичної відповідності. Визначення кількісних показників відповідності веб сервісу та цільового запиту й ранжування сервісів за оцінками ступеня відповідності.

Відповідно до наведеної моделі, оголошений веб сервіс відповідає запиту, якщо:

- контекст сервісу є не меншим ніж контекст запиту;
- передумови оголошеного сервісу є простішими за передумови запиту, або щонайменше їх так само легко задовольнити, як і умови запиту;
- ефекти оголошеного сервісу не менші ніж ефекти запиту. Іншими словами, ефекти, що очікуються (визначаються запитом), повинні задовільнятися в першу чергу;
- сервіс виробляє вихідні параметри, множина яких не менша, ніж множина вихідних параметрів запиту;
- для свого виконання оголошений сервіс вимагає вхідні параметри, множина яких не більша за множину вхідних параметрів запиту.

Формально для сервісу та запиту, представлених розширеною IOPE моделлю, задачу встановлення структурної відповідності можна визначити наступним чином.

*Визначення 1.* Нехай сервіс  $S = \{s | s = (I; O; P; E; D)\}$  з репозиторію  $\mathcal{R}$ , представлений на базі ациклічного TBox  $\mathcal{T}$ , де  $P$  – передумови сервісу,  $I$  – список вхідних параметрів,  $E$  – ефекти (постумови) сервісу,  $O$  – список вихідних параметрів, а запит  $Q$  визначається як бажаний сервіс  $Q = (P'; I'; E'; O'; D')$ , де  $P'$  – передумови,  $I'$  – список вхідних параметрів,  $E'$  – ефекти сервісу,  $O'$  – список вихідних параметрів, то сервіс  $S$  відповідає запиту  $Q$ , якщо  $D' \subseteq D$ ,  $I \subseteq I'$ ,  $O' \subseteq O$  та існує така інтерпретація  $\mathcal{J}$  [5] TBox  $\mathcal{T}$ , що  $P'^{\mathcal{J}} \subseteq P^{\mathcal{J}}$ ,  $E^{\mathcal{J}} \subseteq E'^{\mathcal{J}}$  [6].

На рис. 1 нижче наведено графічну демонстрацію задачі виявлення відповідно до наведеного визначення.

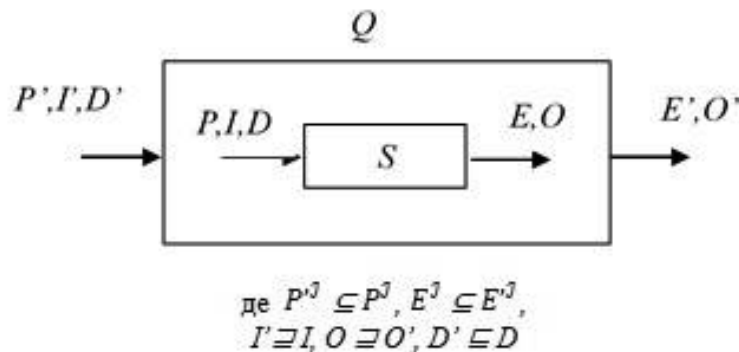


Рис.1. Графічне представлення задачі виявлення

Вирішення задачі виявлення на процесному рівні ускладнюється необхідністю врахування поведінкових аспектів сервісів та внутрішніх семантик самого процесу. Але, слід зазначити, що успішність аналізу процесної моделі цілком залежить від якості її семантичного анотування, головними рисами якої є:

- базування семантичних анотацій на єдиному формальному апараті, який легко піддається автоматизованій обробці (наприклад, онтології, дескриптивні логіки), та уніфікованій системі понять прикладної області;
- анотування всіх елементів та кроків процесу, що є значимими для виконання сервісу, та складають його сутність (тобто повнота системи анотацій);
- відсутність надлишкових анотацій, що перевантажують опис сервісу та ускладнюють його обробку.

Процес сервісу можна розглядати як сукупність значимих для його виконання операцій, кожна з яких має власні характеристики: вхідні та вихідні параметри, умови виконання, та виробляє певні ефекти. Фактично, це атомні сервіси, що можуть залежати або не залежати один від одного, отже, виконуватися послідовно чи паралельно. Ці зв'язки й визначають динаміку сервісу та обумовлюють необхідність не лише порівняння головних складових процесу як атомних сервісів, а й врахування їхніх залежностей у часі. Це вимагає використання для вирішення задач динамічних формалізмів, таких, як, зокрема, темпоральні [7,8] або динамічні [9] дескриптивні логіки.

Пошуковий запит, чи віртуальний бажаний сервіс, з яким здійснюється співставлення, фактично є ціллю, чи цільовим сервісом, отримання якого і є метою вирішення задачі веб сервісів. Цільовий сервіс неформально, являє собою опис бізнес-задачі, яку необхідно розв'язати засобами наявних веб-сервісів. У реальному житті дуже рідко можна знайти один сервіс, що відповідатиме цілі. Тобто, необхідно побудувати складний сервіс із наявних, який буде вирішувати поставлену бізнес-задачу, тобто буде максимально схожим до цільового запиту. В такому разі, задача виявлення зводиться до пошуку сервісів кандидатів на композицію, а задача композиції – до побудови складного сервісу, що відповідає цілі, як ланцюжка сервісів кандидатів на основі їх залежностей та відповідності характеристик. У даному випадку, нагальною є задача визначення відповідностей між сервісами кандидатами, зокрема, відповідності «попередник-послідовник» для впорядкування сервісів у ланцюг.

*Визначення 2.* Маємо множину сервісів кандидатів на композицію  $\mathcal{S}$  та цільовий запит  $Q = \{P_Q; I_Q; E_Q; O_Q; D_Q\}$ , що описуються моделлю (1). Два сервіси  $S1 = \{P_{S1}; I_{S1}; E_{S1}; O_{S1}; D_{S1}\}$  та  $S2 = \{P_{S2}; I_{S2}; E_{S2}; O_{S2}; D_{S2}\}$  можна вважати послідовними  $S1 \rightarrow S2$  в межах результуючого композитного сервісу, якщо:

- контексти обох веб сервісів включаються у контекст цілі:  $D_{S1} \sqsubseteq D_Q$  та  $D_{S2} \sqsubseteq D_Q$ .
- вихідні параметри першого веб сервісу забезпечують входи другого веб сервісу:  $I_{S2} \sqsubseteq O_{S1}$ .
- ефекти, що виробляються першим веб сервісом, забезпечують істинність передумов другого веб сервісу:  $E_{S1}^j \rightarrow P_{S2}^j$ .

Слід зауважити, що дане визначення є обмеженим, адже не враховує складні розгалуження, коли сервіс-послідовник, наприклад, має декілька попередників.

## Онтологічний підхід до вирішення задач веб-сервісів

Онтологічні підходи до вирішення задач передбачають максимальне залучення онтологій на всіх етапах їх вирішення. Це дозволяє досягти уніфікованої строгої формалізації всіх аспектів та забезпечити можливість автоматизованого вирішення із залученням механізмів міркувань. Отже, для вирішення комплексу задач веб сервісів на всіх етапах їхнього життєвого циклу [17] пропонується залучення системи ДЛ онтологій, що містить наступні типи онтологій:

- *Таксономію іменованих сутностей* для категоризації запиту та сервісів при аналізі їхніх контекстних описів [3]. Дана таксономія в дечому схожа на БД, що використовуються для порівняння текстів. Наприклад, на систему класів іменованих сутностей, запропонованих Microsoft [11], або на спеціальні словники понять (наприклад, Wordnet [12]). Але, на відміну від них, враховуючи формалізацію опису сервісу, дана таксономія має сприяти не лише визначенню близькості цільового та наявних описів, а й, перш за все, дозволяти категоризувати наявні сервіси: за областю застосування, типами функцій, що реалізуються, тощо. Тобто вона має охоплювати всі важливі семантичні аспекти контекстних описів сервісів. Її головною метою є визначення максимально повної узагальненої системи понять, що уможливають структурований семантичний опис контексту будь-якого веб-сервісу, незалежно від сфери його застосування чи бізнес-призначення. У [3] пропонується прототип такої ДЛ-онтології (*DocumentationOntology*), визначеної засобами дескриптивної логіки ALC, що забезпечує категоризацію веб сервісів за призначенням, функціональністю, областю знань, тощо.
- *Інтегровану онтологію прикладного домену* - описує предметну область бізнес-задачі, що вирішується. Слід зазначити, що бізнес-задача у загальному випадку, може спиратися на декілька онтологій прикладної області. У такому випадку доцільно їх інтегрувати до загальної онтології домену та використовувати єдиний T-BOX для всіх сервісів, які необхідно анотувати в домені. T-BOX повинен містити всі концепти, які необхідно представляти в домені застосування.
- *Загальну онтологію веб сервісу вищого рівня* – надає загальне визначення веб-сервісу, тобто специфікує його базові сутності, які представляють його головні характеристики. Онтологія сервісу має відображати визначення його моделі, тому її TBox залежить від обраної моделі представлення. ДЛ онтологія веб сервісу, представленого розширеною IOPE моделлю на функціональному рівні, буде наведена нижче.
- *Онтології задач веб-сервісів*. Фактично, є розширеннями загальної онтології веб-сервісу і доповнюють її TBox твердженнями й логічними правилами, які формально описують саме задачу та уможливають автоматизоване прийняття рішення (наприклад, про відповідність оголошення сервісу цілі або про відповідність сервісів один одному у ланцюжку композиції) через застосування резонерів ДЛ.
- *Онтології конкретних веб-сервісів* – дозволяють перекласти оголошення реального сервісу формальною мовою ДЛ. Онтології конкретних сервісів спираються на загальну онтологію сервісу вищого рівня та інтегровану онтологію прикладного домену, а також можуть використовувати сутності з таксономії загальної структури понять (спеціальних словників понять).

Для визначення наведеної системи онтологій попередні дослідження спиралися на базову ДЛ **ALC**, що є ефективним засобом представлення знань та забезпечує виконання стандартних виведень. Але ДЛ-специфікація задач веб сервісів вимагає не лише моделювання «абстрактних» об'єктів (параметра, умови, сервісу тощо), а й «конкретних» їх характеристик, в тому числі характеристик часу і простору. Водночас, ці конкретні величини не можна представити в моделі як елементи області  $\Delta$ , оскільки, переходячи від однієї до іншої інтерпретації, ці «конкретні» елементи та зв'язки між ними будуть змінюватися, тоді як є вимога, щоб вони лишалися не змінними. При використанні логік із конкретними доменами виділяється окрема область з фіксованою множиною предикатів. Тож інтерпретація у новому сенсі складається з інтерпретації у старому сенсі («абстрактної»), що доповнена фіксованою «конкретною» інтерпретацією (множиною з набором предикатів). Очевидно, що це вимагає також використання особливих ролей для зв'язування цих «абстрактних» та «конкретних» елементів конструкторів для побудови концептів, виходячи з таких ролей та «конкретних» предикатів.

Тому, фокусом даного дослідження є ДЛ формалізація задач веб сервісів, зокрема, задачі виявлення та задачі композиції семантичних веб сервісів, за допомогою апарату дескриптивних логік з конкретними доменами.

Перш за все, наведемо основні теоретичні аспекти та синтаксис ДЛ із конкретними доменами, на які будемо спиратися далі у визначенні вказаних онтологій задач.

## Дескриптивні логіки з конкретними доменами. Основні визначення

За своєю сутністю, конкретний домен – це структура певної предикатної сигнатури, тобто множина з заданими на ньому відношеннями. Формально його можна визначити наступним чином.

Визначення 3 [13, 14]. Конкретний домен – це пара  $\mathcal{D}=(D, \Phi)$ , де  $D$  – не пуста множина, а  $\Phi$  – набір предикатів на множині  $D$ . Можна вважати, що задана множина предикатних символів  $PN$ , кожний символ  $P \in PN$  має валентність  $n$ , а  $\Phi$  співставляє йому  $n$ -місне відношення  $P^{\mathcal{D}} \subseteq D^n$ . Окрім цього, будемо враховувати наступні погодження:

- $\mathcal{D}$  містить одномісний предикат, тобто символ  $T$  завжди входить в  $PN$  та інтерпретується як  $T^{\mathcal{D}} = D$ .
- $\mathcal{D}$  є завжди замкненим відносно доповнення, тобто для кожного  $n$ -місного предикатного символу  $P$  в сигнатурі  $PN$  існує  $n$ -місний предикатний символ  $\neg P$ , який інтерпретується як  $\Delta^n / P^{\mathcal{D}}$ .

Зрозуміло, що будь-який домен можна перетворити на область, що задовольняє ці погодження.

На базі наведеного визначення у [15] була введена дескриптивна логіка з конкретним доменом  $\mathcal{ALC}(\mathcal{D})$ .

Визначення 4. Нехай заданий конкретний домен  $\mathcal{D}$  з множиною предикатних символів  $PN$  та наступними кінцевими множинами предикатних символів:  $CN$  – атомарні концепти,  $RN$  – атомарні ролі,  $AF \subseteq RN$  – атомарні абстрактні атрибути,  $CF$  – атомарні конкретні атрибути. Складовим конкретним атрибутом називають ланцюжок  $f_1 \dots f_k h$  з  $k \geq 1$  атомарних абстрактних атрибутів  $f_i \in AF$  та одного конкретного атрибута  $h \in CF$ .

Визначення 5 (Синтаксис логіки  $\mathcal{ALC}(\mathcal{D})$ ). Концепти логіки  $\mathcal{ALC}(\mathcal{D})$  задаються граматикою:

$$T \mid \perp \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists R.C \mid \forall R.C \mid \exists [u_1, \dots, u_n].P,$$

де  $A \in CN$ ,  $R \in RN$ ,  $u_1, \dots, u_n$  – довільні (складні) атрибути,  $P \in PN$  –  $n$ -місний конкретний предикат. Тобто, слід зауважити, що до синтаксису логіки  $\mathcal{ALC}$  додано єдиний новий вид концептів  $\exists [u_1, \dots, u_n].P$ .

Визначення 6 (Семантика логіки  $\mathcal{ALC}(\mathcal{D})$ ) [15]. Інтерпретація  $\mathcal{J} = (\Delta, \mathcal{J})$  визначається аналогічно, як і для логіки  $\mathcal{ALC}$ , але з наступними доповненнями:

- множини  $\Delta$  та  $D$  не перетинаються;
- кожному атомарному абстрактному атрибуту  $f \in AF$  ставиться у відповідність функція  $f^{\mathcal{J}} : \Delta \rightarrow \Delta$ ;
- кожному атомарному конкретному атрибуту  $h \in CF$  ставиться у відповідність функція  $h^{\mathcal{J}} : \Delta \rightarrow D$ ;
- составний конкретний атрибут  $u = f_1 \dots f_k h$  інтерпретується як композиція часткових функцій:  
 $u^{\mathcal{J}}(x) = h^{\mathcal{J}}(f_k^{\mathcal{J}}(\dots f_1^{\mathcal{J}}(\dots)))$ , результатом якої є часткова функція  $u^{\mathcal{J}} : \Delta \rightarrow D$ ;
- новий вид концептів  $\exists [u_1, \dots, u_n].P$  інтерпретується наступним чином:

$$(\exists [u_1, \dots, u_n].P)^{\mathcal{J}} = \{e \in \Delta \mid \exists x_1, \dots, x_n \in D : u_1^{\mathcal{J}}(e) = x_1 \wedge \dots \wedge u_n^{\mathcal{J}}(e) = x_n \wedge (x_1, \dots, x_n) \in P^{\mathcal{D}}\}.$$

Якщо всі функції  $u_i^{\mathcal{J}}$  визначені в точці  $e$ , то наведену формулу можна скоротити:

$$(\exists [u_1, \dots, u_n].P)^{\mathcal{J}} = \{e \in \Delta \mid (u_1^{\mathcal{J}}(e), \dots, u_n^{\mathcal{J}}(e)) \in P^{\mathcal{D}}\}$$

Множина всіх точок, на яких визначений атрибут  $u$ , виражається концептом  $\exists u.T$ , де  $T$  – не концепт, а конкретний предикат. Тоді має місце замкнутість концептів логіки відносно оператора заперечення, тобто:

$$\neg \exists [u_1, \dots, u_n].P \equiv \neg \exists u_1.T \sqcup \dots \sqcup \neg \exists u_n.T \sqcup [u_1, \dots, u_n].\neg P.$$

В [14] розглядаються приклади конкретних доменів, а також питання розв'язуваності відповідних логік та розв'язуваності групи логік з конкретними доменами в цілому й доводиться, що [16]:

- якщо домен  $\mathcal{D}$  є розв'язуваним, то й логіка  $\mathcal{ALC}(\mathcal{D})$  – розв'язувана;
- якщо за складністю домен  $\mathcal{D}$  належить класу  $PSpace$ , то логіка  $\mathcal{ALC}(\mathcal{D})$  є  $PSpace$ -повною [18].

Наведені вище система онтологій та специфікація дескриптивної логіки  $\mathcal{ALC}(\mathcal{D})$  покладені в основу ДЛ-визначення задач виявлення та композиції веб-сервісів.

## Визначення онтологій задач виявлення та композиції веб-сервісів засобами логіки $\mathcal{ALC}(\mathcal{D})$

Як уже було зазначено, онтологія задачі веб-сервісу фактично являє собою набір тверджень ДЛ, що є розширенням загальної онтології сервісу вищого рівня та залежить від обраної моделі представлення веб сервісу. Спираючись на розширену ЮРЕ функціональну модель, визначимо TBox загальної онтології веб сервісу вищого рівня *ServiceOntology* наступним чином (прототип онтології *DocumentationOntology* наведений в [3]).

**T-BOX.** *Service, Condition, Effect, Parameter, Name, Value, Type, Context, DocItem;*

*Service*  $\sqsubseteq$  *has.DocItem*;

*Service*  $\sqsubseteq$  *has.Context*;

*DocItem*  $\sqsubseteq$  *Context*;

*DocItem*  $\sqsubseteq$  *has.ItemName*;

*DocItem*  $\sqsubseteq$  *has.ItemValue*;

*DocItem*  $\sqsubseteq$  *BusinessDomenCategory* / \*зв'язок з онтологією *DocumentationOntology* (визначає категорію бізнес домена)

*DocItem*  $\sqsubseteq$  *BFunctionType* / \* зв'язок з онтологією *DocumentationOntology* (визначає тип функцій, що реалізуються веб-сервісом)

*DocItem*  $\sqsubseteq$  *MainEntitiesTypes* / \* зв'язок з онтологією *DocumentationOntology* (визначає типи сутностей з таксономії іменованих сутностей)

*ItemName*  $\sqsubseteq$  *hasType*.{String};

*ItemValue*  $\sqsubseteq$  *hasType*.Type

*Parameter*  $\sqsubseteq$  *has.Name*;

*Parameter*  $\sqsubseteq$  *has.Value*;

$Name \sqsubseteq hasType.\{String\};$   
 $Value \sqsubseteq hasType.Type;$   
 $Condition \sqsubseteq hasValue.Boolean;$   
 $Service \sqsubseteq hasInputParameters.Parameter; /* I – складова сервісу$   
 $Service \sqsubseteq hasOutputParameters.Parameter /* O – складова сервісу$   
 $Service \sqsubseteq hasPreCondition.Condition; /* P – складова сервісу$   
 $Service \sqsubseteq hasEffect.Effect; /* E – складова сервісу$   
 $Service \sqsubseteq hasServiceContext.Context; /* розширення IOPE – моделі контекстним описом$   
 $Type = \{String, Numeric, Boolean, \dots\}.$

Дане дослідження фокусується на першому етапі специфікації задачі виявлення засобами  $ALC(\mathcal{D})$ , а саме для розширеної IO моделі сервісу, та передбачає подальший розвиток на розширену функціональну IOPE модель веб-сервісу, що наведена вище, а також на процесну модель веб-сервісу.

Визначимо конкретний домен  $\mathcal{D}_{match} = \{\mathcal{S}, \Phi\}$ , де  $\mathcal{S} = \mathcal{S}_P \sqcup \mathcal{S}_C$  є об'єднанням двох наборів:  $\mathcal{S}_P$  – сукупності можливих підмножин множини індивідів концепту *Parameter* та  $\mathcal{S}_C$  – сукупності можливих підмножин множини індивідів концепту *Context*, а  $\Phi$  – набір предикатів, що визначені на  $\mathcal{S}$ . Поки що обмежимо  $\Phi$  одномісними (P1) та двомісними (P2) предикатними символами:  $P1 = \{T\}$ ,  $P2 = \{\subseteq, \supseteq, \cong, \neq, \not\subseteq, \not\supseteq\}$ . Зауважимо, що, враховуючи особливості обраного домена, а саме те, що елементами домена є множини індивідів, за операції на домені обрано теоретико-множинні оператори.

**Онтологія задачі виявлення *MatchingOntology*.** ТВох онтології задачі виявлення *MatchingOntology* на основі логіки  $ALC(\mathcal{D}_{match})$  визначимо наступним чином.

$S \sqsubseteq Service; /*зв'язок з загальною онтологією сервісу верхнього рівня ServiceOntology$   
 $Q \sqsubseteq Service; /*зв'язок з загальною онтологією сервісу верхнього рівня ServiceOntology$

Введемо спеціальний композитний концепт *DiscoveryObject*, який буде виокремлювати абстрактні сервіс-запити з загальної множини наявних у реєстрі веб-сервісів. Фактично, його екземплярами будуть пари сервіс-запит, що відповідають або не відповідають один одному.

$DiscoveryObject \sqsubseteq hasService.Service /*зв'язок з онтологією ServiceOntology;$   
 $DiscoveryObject \sqsubseteq hasQuery.Service; /*зв'язок з онтологією ServiceOntology;$

Нехай *hasAvailableValues* конкретний атрибут в логіці  $ALC(\mathcal{D}_{match})$  зі значеннями в  $\mathcal{S}$ . У такому разі, концепти:  $InputMatching \equiv DiscoveryObject \sqcap (hasService hasInputParameters hasAvailableValues \subseteq hasQuery hasInputParameters hasAvailableValues)$ ,  $OutputMatching \equiv DiscoveryObject \sqcap (hasQuery hasOutputParameters hasAvailableValues \subseteq hasService hasOutputParameters hasAvailableValues)$  та  $ContextMatching \equiv DiscoveryObject \sqcap (hasQuery hasContext hasAvailableValues \subseteq hasService hasContext hasAvailableValues)$  визначають множини об'єктів, які описують пари сервіс – запит, відповідних за вхідними/вихідними параметрами. Одразу слід зазначити, що за наданого визначення концептів та домену всі наведені ланцюжки використовують тільки функціональні ролі.

Концепт  $DiscoveryServices \equiv InputMatching \sqcap OutputMatching \sqcap ContextMatching$  визначатиме множини об'єктів, які описують пари сервіс – запит, відповідних за вхідними, вихідними параметрами та контекстами одночасно.

**Онтологія задачі композиції *CompositionOntology*.** ТВох онтології задачі композиції *CompositionOntology* на основі логіки  $ALC(\mathcal{D}_{match})$  визначимо наступним чином.

$S1 \sqsubseteq Service; /*зв'язок з загальною онтологією сервісу верхнього рівня ServiceOntology$   
 $S2 \sqsubseteq Service; /*зв'язок з загальною онтологією сервісу верхнього рівня ServiceOntology$   
 $Q \sqsubseteq Service; /*зв'язок з загальною онтологією сервісу верхнього рівня ServiceOntology$

Введемо спеціальний композитний концепт *CompositionObject*, екземплярами якого будуть пари сервіс-сервіс, що відповідають або не відповідають один одному.

$CompositionObject \sqsubseteq hasService1.Service /*зв'язок з онтологією ServiceOntology;$   
 $CompositionObject \sqsubseteq hasService2.Service; /*зв'язок з онтологією ServiceOntology;$   
 $CompositionObject \sqsubseteq hasQuery.Service; /*зв'язок з онтологією ServiceOntology;$

Нехай *hasAvailableValues* конкретний атрибут у логіці  $ALC(\mathcal{D}_{match})$  зі значеннями в  $\mathcal{S}$ . У такому разі, концепт  $ServiceMatching \equiv CompositionObject \sqcap (hasService2 hasInputParameters hasAvailableValues \subseteq hasService1 hasOutputParameters hasAvailableValues) \sqcap (hasService2 hasContext hasAvailableValues \subseteq hasQuery hasContext hasAvailableValues) \sqcap (hasService1 hasContext hasAvailableValues \subseteq hasQuery hasContext hasAvailableValues)$  визначає множини об'єктів, які описують пари сервіс – сервіс, відповідних один одному за вхідними/вихідними параметрами, тобто з сервісів  $S1$  та  $S2$  можна побудувати композитний сервіс як їх послідовний ланцюжок  $S1 \rightarrow S2$ . Зауважимо, що контексти обох сервісів мають включатися до контексту запиту. Ця умова може виглядати надто суворою. Насправді, може бути достатньою наявність не порожнього перетину множин значень контекстів сервісів та цільового запиту.

Подальший розвиток наведених онтологій задач пов'язаний із введенням концептів, які будуть визначати пари сервіс – запит, що є схожими відповідно за контекстами, передумовами та ефектами, які надалі в свою чергу, повинні розширити визначення складних концептів *DiscoveryServices* та *CompositionObject*, вдосконалюючи визначення задач додатковими характеристиками відповідності веб-сервісів. Окрім цього, слід зауважити, що ланцюжок композиції реально набагато ширше за послідовність сервісів, сумісних за входами/виходами, та може містити розгалуження, визначення яких також передбачає подальший розвиток онтології *CompositionObject*.

## Висновки

Дане дослідження було спрямоване на створення ефективних розв'язків проблеми співставлення веб-сервісів у межах вирішення головних задач веб-сервісів, а саме задач їх виявлення та композиції. В основу запропонованого підходу було покладено ідею максимальної онтологізації задачі, що передбачає використання ДЛ онтологій як для опису самих сервісів та цілі, так і для представлення задач, процесів та алгоритмів. В межах підходу були визначені п'ять головних типів онтологій, а саме: (1) таксономія іменованих сутностей, (2) загальна онтологія сервісу верхнього рівня, (3) уніфікована онтологія прикладного домена, (4) онтології задач веб-сервісів та (5) онтології конкретних сервісів. Набір перелічених онтологій складає єдину інформаційну систему, всі онтології є пов'язаними. Так, онтології конкретних сервісів містять зв'язки з онтологіями всіх чотирьох інших типів. Для представлення всіх зазначених онтологій було використано апарат дескриптивних логік. В основу вирішення проблеми у даному дослідженні було покладено використання дескриптивних логік із конкретними доменами. Зокрема, було запропоновано логіку  $ALC(\mathcal{D}_{match})$ , де конкретний домен специфікується набором усіх підмножин множини індивідів концепта *Parameter*, що охоплює всі можливі параметри наявних та бажаних веб-сервісів з теоретико-множинними операціями на ньому. Застосування  $ALC(\mathcal{D}_{match})$ , визначеної таким чином, дозволяє представити основні характеристики сервісу функціональними ролями дескриптивної логіки, а також на їхній основі побудувати концепти, екземпляри яких фактично уможливають вирішення поставлених задач. Отримані рішення є складними концептами, побудованими з ланцюжків функціональних ролей онтологій, конкретних атрибутів концептів, теоретико-множинних операторів та базових операторів дескриптивних логік. Так, концепт *DiscoveryServices* визначає пари сервісів (сервіс-запит), відповідні один одному за входами та виходами, а концепт *CompositionObject* – пари сервісів (сервіс-сервіс), що зв'язані відповідністю «попередник-послідовник». Враховуючи високу ступінь можливостей формалізації описів та механізми міркувань, що надаються апаратом дескриптивних логік, даний підхід забезпечує можливість автоматизованого ефективного вирішення вказаних задач.

Але, слід зазначити, що запропоновані в роботі онтології задач є обмеженими, тому що:

(1) вони пропонують рішення на спрощеній моделі представлення сервісів, охоплюючи співставлення лише вхідних та вихідних параметрів;

(2) для задачі виявлення розглядається повна відповідність. Для формування наборів сервісів-кандидатів, що покривають цільовий запит, необхідно полегшити умови відповідності та розглядати варіанти декомпозиції цілі;

(3) для задачі композиції не включені варіанти складних розгалужень та наявності декількох попередників, які забезпечують виконання сервіса-послідовника.

Але наведений підхід є відкритим, а відповідні онтології задач передбачають розширення для формування розв'язків задач із поступовим зняттям вказаних обмежень, що є предметом подальших досліджень.

Ще один важливий аспект – це оцінка розв'язуваності логіки  $ALC(\mathcal{D}_{match})$ , яка фактично зводиться до визначення розв'язуваності домена  $\mathcal{D}_{match}$ , що також є предметом подальших досліджень.

## Література

1. Baader, F., Lutz, C., Milieie, M., Sattler, U., Wolter, F., (2005) A Description Logic Based Approach to Reasoning about Web Services. *In Proceedings of the WWW 2005 Workshop on Web Service Semantics (WSS2005)*
2. Baader, F., Lutz, C., Milieie, M., Sattler, U., Wolter, F., (2005) Integrating Description Logics and Action Formalisms for Reasoning about Web-services. *Technical Report, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology*, volume LTCS-05-02, LTCS-Report 05-02
3. Захарова, О. (2020) Контекстне співставлення веб-сервісів у вирішенні задач виявлення. Онтологічні підходи. *Проблеми програмування*. 2020. № 2-3. С.39-49
4. Захарова, О. (2017) Визначення та вирішення задач виявлення веб-сервісів за допомогою апарату дескриптивних логік. *Проблеми програмування*. 2017. № 4. С. 66–78.
5. Staab, S. & Studer, R. (2009) *Handbook on Ontologies*. International Handbooks on Information Systems (INFOSYS). Second edition.
6. Lutz, C. & Sattler, U., (2002) A Proposal for Describing Services with DLs. *In: Int. Workshop on Description Logics*
7. Emerson, E. (1990) Temporal and Modal Logic J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Volume B: Formal Models and Semantics, pages 995–1072, Elsevier, 1990.
8. Hao, S. & Zhang, L. (2010) Dynamic Web Services Composition Based on Linear Temporal Logic. Conference: *Information Science and Management Engineering (ISME)*, Volume: 1
9. (PDF) *A Dynamic Description Logic for Representation and Reasoning About Actions* (researchgate.net)
10. Chang, L., Lin, F., Shi, Z., (2007) A dynamic description logic for representation and reasoning about actions. *KSEM'07: Proceedings of the 2nd international conference on Knowledge science, engineering and management*. November 2007 P. 115–127
11. Supported Categories for Named Entity Recognition - Azure Cognitive Services. *Microsoft Docs*
12. Peng, Y. (2010) Two levels semantic web service discovery. *In seventh international conference on fuzzy systems and knowledge discovery*, Aug. 2010.
13. Lutz, C. (2003) Description Logics with Concrete Domains - A Survey. *Advances in Modal Logics*, Volume 4. King's College Publications, 2003.
14. Zolin, E. Description logic. Special course lectures at [logic.math.msu.ru/staff/zolin/dl](http://logic.math.msu.ru/staff/zolin/dl).
15. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (2005) *The Description Logic Handbook*. Cambridge University Press, 2003.
16. Baader, F. & Hanske, P. (1991) A schema for integrating concrete domains into concept languages. *In Proceeding of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, P. 452-457, Sydney.
17. Захарова, О. (2018) Методика застосування апарату дескриптивних логік у процесі побудови композитного сервісу на функціональному рівні. *Проблеми програмування*. 2018. №1
18. <https://ru-frwiki.wiki/wiki/PSPACE>

## References

1. Baader, F., Lutz, C., Milieie, M., Sattler, U., Wolter, F., (2005) A Description Logic Based Approach to Reasoning about Web Services. In *Proceedings of the WWW 2005 Workshop on Web Service Semantics (WSS2005)*
2. Baader, F., Lutz, C., Milieie, M., Sattler, U., Wolter, F., (2005) Integrating Description Logics and Action Formalisms for Reasoning about Web-services. *Technical Report, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology*, volume LTCS-05-02, LTCS-Report 05-02
3. Zakharova, O. (2020) Context web services matching in the discovery task resolving. Ontological approaches. *Problems in programming*. 2020. № 2-3. P. 39-49
4. Zakharova, O. (2017) Defining and resolving Web-services discovery problems using description logics formalism. *Problems in programming*. 2017. № 4. P. 66–78.
5. Staab, S. & Studer, R. (2009) *Handbook on Ontologies*. International Handbooks on Information Systems (INFOSYS). Second edition.
6. Lutz, C. & Sattler, U., (2002) A Proposal for Describing Services with DLs. In: *Int. Workshop on Description Logics*
7. Emerson, E. (1990) Temporal and Modal Logic J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, Volume B: Formal Models and Semantics, pages 995–1072, Elsevier, 1990.
8. Hao, S. & Zhang, L. (2010) Dynamic Web Services Composition Based on Linear Temporal Logic. Conference: *Information Science and Management Engineering (ISME)*, Volume: 1
9. [\(PDF\) A Dynamic Description Logic for Representation and Reasoning About Actions \(researchgate.net\)](#).
10. Chang, L., Lin, F., Shi, Z., (2007) A dynamic description logic for representation and reasoning about actions. *KSEM'07: Proceedings of the 2nd international conference on Knowledge science, engineering and management*. November 2007 P. 115–127
11. Supported Categories for Named Entity Recognition - Azure Cognitive Services. *Microsoft Docs*
12. Peng, Y. (2010) Two levels semantic web service discovery. In *seventh international conference on fuzzy systems and knowledge discovery*, Aug. 2010.
13. Lutz, C. (2003) Description Logics with Concrete Domains - A Survey. *Advances in Modal Logics*, Volume 4. King's College Publications, 2003.
14. Zolin, E. Description logic. Special course lectures at [logic.math.msu.ru/staff/zolin/dl](http://logic.math.msu.ru/staff/zolin/dl).
15. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (2005) *The Description Logic Handbook*. Cambridge University Press, 2003.
16. Baader, F. & Hanske, P. (1991) A schema for integrating concrete domains into concept languages. In *Proceeding of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, P. 452-457, Sydney.
17. Zakharova, O. (2018) The technique of using Description Logics in the process of constructing a composite service at the functional level. *Problems in programming*. 2018. № 1. P. 77-91.
18. <https://ru.frwiki.wiki/wiki/PSPACE>

Одержано 01.08.2022

### Про автора:

Захарова Ольга Вікторівна,

кандидат технічних наук,

старший науковий співробітник.

Кількість наукових публікацій в українських виданнях – 32.

<http://orcid.org/0000-0002-9579-2973>.

### Місце роботи автора:

Інститут програмних систем НАН України,

проспект Академіка Глушкова, 40.

E-mail: [ozakharova68@gmail.com](mailto:ozakharova68@gmail.com).

Моб.тел.: +38(068)594756

### Прізвища та ім'я авторів і назва доповіді англійською мовою:

Zakharova O. V.

Applying Description Logics with Concrete Domains to Solve the Problems of Semantic Web Services Discovery and Composition

### Прізвища та ім'я авторів і назва доповіді українською мовою:

Захарова О. В.

Застосування дескриптивних логік з конкретними доменами

до вирішення задач виявлення та композиції семантичних веб сервісів

**Контакти для редактора:** Захарова Ольга Вікторівна, старший науковий співробітник Інституту програмних систем НАН України, e-mail: [ozakharova68@gmail.com](mailto:ozakharova68@gmail.com), тел.: (38)(068) 594-75-60