

БАЗИ ДАНИХ, БАЗИ ЗНАТЬ ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.94

Захарова О.В.

Інститут програмних систем НАН України

ПІДХОДИ ДО ВИРІШЕННЯ ПРОБЛЕМИ ЕФЕКТИВНОСТІ ВИЯВЛЕННЯ СЕМАНТИЧНИХ ВЕБ- СЕРВІСІВ НА ФУНКЦІОНАЛЬНОМУ РІВНІ

Метою дослідження є аналіз можливих підходів до автоматичного виявлення сервісів, що задовольняли б пошуковому запиту. Опис запиту по суті є описом сервісу, що запитується. Наша ціль – за допомогою кон'юнкцій описів існуючих сервісів побудувати такий сервіс, що запитується. Малоімовірно, що можливо побудувати сервіс, еквівалентний до запиту, тому необхідно побудувати його найкраще покриття, де під «найкращим» розуміється «найменше». Розглянута в роботі задача найкращого покриття є одним з екземплярів задач ререйтингу, де в якості відношення між запитом та покриттям використовується семантична різниця між концептами замість включення або еквівалентності. Точніше, задача зводиться до того, щоб переформулювати запит Q в опис, що містить якнайбільше інформації, загальної з Q . Це переформулювання і є покриттям Q .

Цель исследований заключается в анализе возможных подходов к автоматическому обнаружению сервисов, удовлетворяющих поисковому запросу. Описание запроса, по сути, является описанием запрашиваемого сервиса. Наша цель – посредством конъюнкций описаний существующих сервисов построить запрашиваемый сервис. Маловероятна возможность построения сервиса, эквивалентного запросу, поэтому необходимо построить его наилучшее покрытие, где под «наилучшим» понимается «наименьшее». Рассмотренная в работе задача наилучшего покрытия является одним из экземпляров задач ререйтинга, где в качестве отношения между запросом и покрытием используется семантическая разность между концептами вместо включения или эквивалентности. Точнее, задача сводится к переформулированию запроса Q в описание, содержащее максимально возможное количество общей с Q информации. Это переформулирование и является покрытием Q .

Purpose of research is in analysis of possible approaches to automatic discovery Web-services satisfied to the query. In fact, query description is description of requested service. Our purpose is to build requested service with conjunction of existed services. It is unlikely to build service that is equivalent to query, so we need to create its best cover where “the best” means “the least”. Considered the best covering problem is one instance of the general framework for rewriting that use semantic distance between concepts, instead of subsumption or equivalence. Precisely, the problem reduces to reformulate a query Q into a description that contain as much as possible of common information with Q . Such reformulation is a cover of Q .

Ключові слова: семантичний веб, веб-сервіс, виявлення веб-сервісів, дескриптивна логіка, найкраще покриття запиту.

Вступ

Семантичні Веб-сервіси є об'єднанням Семантичного Веба та сервіс-орієнтованих обчислень. Така комбінація забезпечує засновану на веб інфраструктуру для загального обміну знаннями та повторного використання. Семантичні Веб-сервіси дозволяють об'єднати джерела знань у вигляді Веб-сервісів та описати взаємодію та робочі процеси між ними за допомогою XML [1].

Семантичні Веб-сервіси створюються та розвиваються як перспективні технології для ефективної автоматизації процесів виявлення сервісів, їх композиції та керування сервісами [2,3,4]. Критерієм відбору Веб-сервісів може бути відповідність їх можливостей пошуковому запиту. Порівняння сервісів та запиту здійснюється на основі їх вхідних та вихідних параметрів, передумов та після-умов. Для встановлення відповідності необхідні

сучасні ефективні алгоритми метченгу (matching).

Один з підходів до ефективного вирішення задачі виявлення Веб-сервісів на базі їх можливостей є задача знаходження найкращого покриття, що використовує апарат дескриптивних логік.

Загальні означення

Семантичний Веб – це структура для створення універсального середовища для обміну інформацією, пов'язуючи семантики з документами в World Wide Web.

Сервіс-орієнтовані обчислення – це архітектура, що дозволяє, щоб інформаційні ресурси були представлені як незалежні від платформи, само-описувані, модульні програмні компоненти.

Сервіс – це шістка його передумов, вхідних параметрів, ефектів, об'єктів, що виробляються в якості ефектів, вихідних параметрів та після умов. $S = (CI; I; A; AO; O; CO)$ є представленням сервісу, де CI – передумови сервісу, I – список вхідних параметрів, A – ефекти сервісу, AO – об'єкти, що виробляються як ефекти сервісу, O – список вихідних параметрів та CO – після-умови сервісу.

Репозиторій Веб-сервісів – це множина Веб-сервісів.

Запит сервісу [5] визначається як $Q = (CI'; I'; A'; AO'; O'; CO')$, де CI' – передумови, I' – список вхідних параметрів, A' – ефекти сервісу, AO' – об'єкт ефекту, O' – список вихідних параметрів та CO' – після-умови. Всі ці елементи є параметрами сервісу, що запитується.

Формальне визначення задачі виявлення Веб-сервісів

Припустимо, що задано репозиторій Веб-сервісів, та запит, що запитує сервіс, тоді автоматичне знаходження в репозиторії сервісу, що відповідає запиту є Задачею виявлення Веб-сервісу [5]. Валідні рішення щодо запиту задовольняють наступні умови: (i) вони виробляють хоча б один вихідний параметр запиту та задовольняють всі передумови запиту; (ii) вони використовують вхідні параметри тільки з наданого списку вхідних параметрів та задовольняють передумови запиту; (iii) вони виробляють ефекти запиту. Деякі рішення можуть бути занадто обмеженими, але вони все одно розглядаються як дійсні, якщо задовольняють вимогам вхідних та вихідних параметрів, перед/пост умов та ефектів.

Наприклад, нам треба знайти сервіс покупки книжок, директорія сервісів містить сервіси S_1 та S_2 (табл.):

Таблиця
Сервіси та їх характеристики

Сервіс	Входи	Передумови	Виходи	Пост-умови
S_1	<i>BookName,</i> <i>AuthorName,</i> <i>BookISBN,</i> <i>CreditCardNumber</i>		<i>ConfirmationNumber</i>	-
S_2	<i>BookName,</i> <i>CreditCardNumber</i>	<i>IsNumeric(CreditCardNumber)</i>	<i>ConfirmationNumber,</i> <i>TrackingNumber</i>	-
Запит				-
Q	<i>BookTitle,</i> <i>CreditCardNumber,</i> <i>AuthorName,</i> <i>CreditCardType</i>	<i>IsNumeric(CreditCardNumber)</i>	<i>ConfirmationNumber</i>	-

Сервіс S_1 має вхідні параметри *BookName*, *AuthorName*, *BookISBN*, *CreditCardNumber* та вихідний параметр *ConfirmationNumber*. Сервіс S_2 має вхідні параметри *BookName* та *CreditCardNumber*, вихідні параметри *ConfirmationNumber* і *TrackingNumber* та для його виконання задана передумова *IsNumeric(CreditCardNumber)*. Для пошуку сервісу покупки книжок заданий запит зі

вхідними параметрами *BookTitle*, *CreditCardNumber*, *AuthorName*, *CreditCardType*, вихідним параметром *ConfirmationNumber* та передумовою на номер кредитної картки *IsNumeric(CreditCardNumber)*.

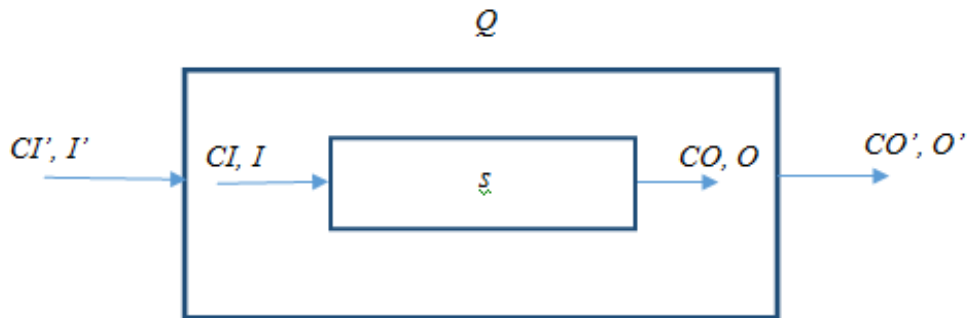
Семантичні описи вхідних та вихідних параметрів сервісу повинні бути такі самі як для параметрів запиту або мати відношення

subsumption. Механізм виявлення повинен мати можливість вивести той факт, що параметр запиту *BookTitle* та вхідний параметр *BookName* сервісу S_2 семантично одні й ті самі концепти. Це може бути виведено, використовуючи семантики онтології домена. Запит також має передумову на *CreditCardNumber*, яке повинно мати числове значення, що логічно повинно означати передумову сервісу, який шукається.

Означення. Заданий репозиторій \mathcal{R} та запит Q , задача виявлення може бути визначена як автоматичне знаходження множини сервісів S з репозиторію \mathcal{R} таких, що $S = \{s/ s = (CI; I; A; AO; O; CO), s \in R, CI' \Rightarrow CI, I \subseteq I', A = A', AO = AO', CO \Rightarrow CO', O \supseteq O'\}$. \subseteq означає відношення

включення (subsumes), а \Rightarrow – відношення імплікації.

Наприклад, нехай x та y вхідні та вихідні параметри сервісу, відповідно. Якщо запит містить умову ($x > 5$), як передумову, та ($y > -x$), як після-умову, то сервіс з передумовою ($x > 0$) та після-умовою ($y > x$) може задовольняти запит, тому що ($x > 5$) \Rightarrow ($x > 0$) та ($y > x$) \Rightarrow ($y > -x$), так як ($x > 0$). Іншими словами, задача виявлення втягує знаходження сервісів з репозиторію, що відповідають (match) вимогам запита. Валідні рішення повинні виробляти хоча б один з вихідних параметрів, що специфіковані у запиті, задовольняти перед- та після-умови, використовувати хоча б ті вхідні параметри, які забезпечуються запитом, та виробляти такий самий ефект, як вимагається запитом. На рис. графічно пояснюється задача виявлення.



де $CI' \Rightarrow CI, CO \Rightarrow CO', I' \supseteq I, O \supseteq O'$

Рис. Виявлення множини сервісів з репозиторію

Дескриптивна логіка

Дескриптивні логіки (ДЛ) [6] дозволяють представити домен, що нас цікавить, у термінах концептів або описів (унарні предикати), що характеризують підмножини об'єктів (екземплярів) в домені, та ролей (бінарні предикати) на такому домені. Концепти визначаються виразами, які формуються за допомогою спеціальних конструкторів [6]: верхній концепт (\top), нижній концепт (\perp), кон'юнкція концептів (Π), універсальний кваліфікатор ($\forall R.C$), числові обмеження ($\geq nR$) та ($\leq nR$) (для різних ДЛ цей набір конструкторів різниться).

Так для наведеного вище прикладу TBox можна визначити наступним чином:

BookName, AuthorName, BookISBN, CreditCardNumber, Type, ConfirmationNumber, TrackingNumber, CreditCardType, Operation, Service, I, O, CI, CO, hasProperty(Service, I),

*hasProperty(Service, O),
hasProperty(Service, CI),
hasProperty(Service, CO),
hasType(CreditCardNumber, Type),
hasISBN(BookName, BookISBN),
hasAuthor(BookName, AuthorName),
hasCreditCard(AuthorName, CreditCardNumber),
hasCardType(CreditCardNumber, CreditCardType),
hasConfirmation(Operation, ConfirmationNumber),
hasTNumber(Operation, TrackingNumber),
BookTitle \equiv BookName
ABox
Type(Numeric)*

В такому випадку запит можна визначити як:

$I(Q) \equiv hasProperty.BookTitle \Pi hasProperty.CreditCardNumber \Pi hasProperty.AuthorName \Pi hasProperty.CreditCardType$
 $O(Q) \equiv hasProperty.ConfirmationNumber$

$CI(Q) \equiv CreditCardNumber \Pi$
 $\exists hasType.Numeric$
 $CreditCardNumber \subseteq \exists hasType.Numeric$
 Сервіс $S1$ можна визначити як:
 $I(S1) \equiv hasProperty.BookName \Pi$
 $hasProperty.AthorName \Pi hasProperty.BookISBN$
 $N \Pi hasProperty.CreditCardNumber$
 $O(S1) \equiv ConfirmationNumber$
 Сервіс $S2$ можна визначити як:
 $I(S2) \equiv hasProperty.BookName \Pi$
 $hasProperty.CreditCardNumber$
 $O(S2) \equiv hasProperty.ConfirmationNumber$
 $\Pi hasProperty.TrackingNumber$
 $CI(S2) \equiv CreditCardNumber \Pi$
 $\exists hasType.Numeric$

Враховуючи аксіому еквівалентності $BookTitle \equiv BookName$, можемо встановити наступні співвідношення між елементами кортежів описів сервісів та запиту.

Для сервісу $S2$:

$I(S2) \subseteq I(Q)$ – для входів,

$O(S2) \supseteq O(Q)$ – для виходів,

$CI(Q) \Rightarrow CI(S2)$, в силу їх еквівалентності

– для передумов.

Для сервісу $S1$:

$I(S1) \subseteq I(Q)$ – для входів,

$O(S1) \supseteq O(Q)$ та $O(S1) \supseteq O(Q)$ – для

виходів.

Таким чином, сервіс $S2$ задовольняє запиту, а сервіс $S1$ – ні, тому що він вимагає в якості вхідного параметра $BookISBN$, що не забезпечується запитом. Запит вимагає $ConfirmationNumber$ в якості вихідного параметра та $S2$ виробляє $ConfirmationNumber$ та $TrackingNumber$. Додатковий вихідний параметр може просто ігноруватися.

Наведений приклад розглядає задачу встановлення відповідності окремого сервісу з репозиторію та запиту, але в реальності малоімовірно знайти один існуючий сервіс, що буде задовольняти запиту для вирішення задачі. Як правило, нам треба знайти множину сервісів, що разом дозволять нам визначити новий сервіс, який буде відповідати запиту, або іншими словами, «покривати» цей запит.

За умови представлення сервісів засобами ДЛ, кожний елемент визначення сервісу та запиту є концептом (класом), визначенням якого є його опис.

Семантики описів концептів визначаються в термінах інтерпретації $\mathfrak{I} = (\Delta^{\mathfrak{I}}, \mathfrak{I})$, що складається з не пустої множини $\Delta^{\mathfrak{I}}$ – інтерпретації домена та функції інтерпретації \mathfrak{I} , яка зв'язує з кожним іменем концепта

$P \in C$ підмножину $P^{\mathfrak{I}}$ of $\Delta^{\mathfrak{I}}$ та з кожним іменем ролі $R \in \mathfrak{R}$ бінарне відношення $R^{\mathfrak{I}} \subseteq \Delta^{\mathfrak{I}} \times \Delta^{\mathfrak{I}}$. На основі цієї семантики визначаються subsumption (включення), еквівалентність та поняття найменшого загального включення (lcs) [7].

Нехай $C, C1, \dots, Cn$ та D – описи концептів:

– C включається в (subsumed by) D ($C \subseteq D$), якщо $C^{\mathfrak{I}} \subseteq D^{\mathfrak{I}}$ для всіх інтерпретацій \mathfrak{I} .

– C еквівалентно D ($C \equiv D$), якщо $C^{\mathfrak{I}} = D^{\mathfrak{I}}$ для всіх інтерпретацій \mathfrak{I} .

– D є найменшим загальним включенням концептів $C1, \dots, Cn$ ($D = lcs(C1, \dots, Cn)$), якщо:

(1) $Ci \subseteq D$ для всіх $1 \leq i \leq n$, та

(2) D найменший опис концепта, що має таку властивість, тобто, якщо D' – опис концепта, що задовольняє $Ci \subseteq D'$ для всіх $1 \leq i \leq n$, тоді $D \subseteq D'$ [7].

Таким чином, найменше загальне включення множини концептів відповідає найбільш специфічному опису, що містить всі концепти з множини. Більш детально питання побудови найменшого загального включення висвітлюється в [13].

Внутрішні описи, що містяться у базі знань, побудовані з використанням ДЛ, називаються термінологією.

Нехай A – ім'я концепта, а C – опис концепта. Тоді $A = C$ – визначення концепта.

Термінологія T – це кінцева множина визначень концептів таких, що кожне ім'я концепта зустрічається в лівій частині визначення не більше одного разу.

Ім'я концепта A називається визначеним концептом у термінології T , якщо воно зустрічається в лівій частині визначення концепта в T . Іншими словами, A називається атомним концептом. Інтерпретація \mathfrak{I} задовольняє твердження $A = C$, якщо $A^{\mathfrak{I}} = C^{\mathfrak{I}}$. Інтерпретація \mathfrak{I} є моделлю для термінології T , якщо \mathfrak{I} задовольняє всім твердженням в T .

Формальна постановка задачі знаходження найкращого покриття

Процес виявлення сервісів можна розглядати як процес перезапису (rewriting), в якому запит Q перезаписується найближчим до нього описом E , який виражається кон'юнкцією Веб-сервісів заданої онтології T (онтології, на базі якої визначаються сервіси та запит). Такий опис і є найкращим покриттям запиту. Слід зазначити, що, враховуючи

низьку ймовірність знаходження сервісу, повністю еквівалентного запиту, для відбору множини сервісів, які будуть складати покриття, потрібний гнучкий механізм співставлення. Таку гнучкість може забезпечити використання операції різниці на описах сервісів. Використання операції різниці дає можливість витягти з підмножини описів Веб-сервісів частини, що є семантично загальною із сервісом, що запитується, та частини, що семантично відрізняється від запиту. Знання цих частин дозволить вибрати релевантні Веб-сервіси, та згодом обрати один найкращий.

Формально операцію різниці можна визначити наступним чином.

Нехай C, D – два описи концептів та $C \subseteq D$.

Різність [8] $C-D$ описів C та D визначається наступним чином: $C-D := \max\{B/B \Pi D \equiv C\}$. Різність двох описів концептів C та D визначається як опис, який містить всю інформацію, яка є частиною C , але не є частиною опису D . Це визначення операції різності вимагає, щоб другий операнд включався в перший. Але у випадку, коли операнди C та D не можуть бути порівняні відносно відношення включення, різність $C-D$ може задаватися через побудову найменшого загального включення C та D , тобто, $C-D := C - lcs(C, D)$. Слід зазначити, що в деяких ДЛ, множина $C-D$ може містити описи, які не є семантично еквівалентними.

Якщо в логіці всі описи в $C-D$ семантично еквівалентні, то різниця є семантично унікальною, та може бути реалізована як операція на описах. Якщо в логіці $C-D$ може містити описи, що не є семантично еквівалентними, різниця не є вірною операцією на описах, та є лише відношенням. В багатьох прикладних системах будь-які окремі максимальні описи з множини є однаково корисними в якості опису різниці. Це завдяки тому факту, що кожний член $C-D$ покриває деяким чином інформаційну різницю між C та D . Якщо цього достатньо, то операція різниці може бути реалізована як недерміністична операція на описах, іншими словами, як багатозначна операція.

Слід зазначити, що з включення $C \subseteq D$ виходить, що C та D повинні бути одного й того самого типу. А так як кон'юнкції визначаються лише на описах одного типу, то результат різниці B повинен також мати той самий тип. Таким чином, різниця двох концептів – це множина концептів, а різниця двох властивостей – це множина властивостей.

Нехай Λ – ДЛ.

– Клозом в Λ є опис A з наступними властивостями: $(A \equiv B \Pi A) \Rightarrow (B \equiv T) \vee (B \equiv A)$. Кожна кон'юнкція $A1 \Pi \dots \Pi An$ клозів може бути представлена множиною клозів $\{A1, \dots, An\}$.

– $A = \{A1, \dots, An\}$ називається скороченою множиною клозів, якщо або $n = 1$, або жоден клоз не включає кон'юнкцію інших клозів: $\forall 1 \leq i \leq n : A \supseteq A \setminus Ai$. Тоді, множина A називається скороченою формою клоза (RCF) кожного опису $B \equiv A1 \Pi \dots \Pi An$.

– Нехай $A = \{A1, \dots, An\}$ та $B = \{B1, \dots, Bm\}$ будуть скороченими множинами клозів у ДЛ Λ . A та B є еквівалентними ($A \equiv B$), якщо $n = m \wedge \forall 1 \leq i \leq n \exists 1 \leq j, k \leq n : Ai \equiv Bj \wedge Bi \equiv Ak$.

– Якщо в ДЛ для кожного опису всі його RCF є еквівалентними структурами, кажуть, що RCF є структурно унікальними в цій логіці.

Операція структурної різниці визначається як множинна різниця між множинами клозів, де клози порівнюються на базі відношення еквівалентності.

Кажуть, що відношення включення в ДЛ Λ є структурним [8], якщо для будь-якого клоза $A \in \Lambda$ та будь-якого опису $B = B1 \Pi \dots \Pi Bm \in \Lambda$, що задається своїм RCF, виконується: $A \equiv B \Leftrightarrow \exists 1 \leq i \leq m : A \supseteq Bi$.

Слід зазначити [8], що:

1) у ДЛ зі структурно унікальними RCF, операція різності може бути визначена безпосередньо використовуючи операцію структурної різниці, та

2) структурне включення є достатньою умовою для ДЛ, щоб мати структурно унікальний RCF. Отже, структурне включення є достатньою умовою, що дозволяє ідентифікувати логіки, де операція різності є семантично унікальною та може бути реалізована використовуючи операцію структурної різності.

Розмір опису. Нехай Λ – ДЛ зі структурним включенням. Розмір $|C|$ опису концепта C Λ -логіки – це число клозів в його RCF. Якщо необхідно, можна прийняти до уваги розмір кожного клозу.

Нехай Λ - ДЛ зі структурним включенням, T - Λ -термінологія та $Q \neq \perp$ когерентний опис Λ -концепта. Множина концептів, які зустрічаються в T , позначається як $S_T = \{Si, i \in [1, n]\}$ з $Si \neq \perp, \forall i \in [1, n]$. Далі, вважаємо, що описи концептів $Si, i \in [1, n]$ представляються їх RCF.

Покриття Q , що використовує термінологію T , є кон'юнкцією ε деяких імен

S_i з T таких, що: $Q - \text{Ics}_T(Q, \varepsilon) \neq Q$. Тобто, покриття концепта Q , що використовує T , визначається як будь-яка кон'юнкція концептів, що зустрічаються в T , які поділяють деяку загальну інформацію з Q . Слід зазначити, що покриття ε опису концепта Q завжди погоджено з Q (тобто $Q \cap \varepsilon \neq \perp$), так як Λ – ДЛ зі структурно унікальним RCF та ми маємо $Q \neq \perp$ та $S_i \neq \perp, \forall i \in [1, n]$.

Щоб формально визначити найкраще покриття, нам спочатку необхідно визначити частину опису покриття ε , яка не міститься в описі запита Q , та частину запиту Q , що не міститься в описі його покриття ε .

Нехай Q – опис Λ -концепту та ε – покриття Q , що використовує T .

Залишок (Решта частина) Q [5] відносно ε , позначається як $Rest_\varepsilon Q$, визначається наступним чином: $Rest_\varepsilon Q \equiv Q - \text{Ics}_T(Q, \varepsilon)$.

Відсутня інформація [5] Q відносно ε , позначається як $Miss_\varepsilon Q$, визначається наступним чином: $Miss_\varepsilon Q \equiv \varepsilon - \text{Ics}_T(Q, \varepsilon)$.

Опис концепта ε називається найкращим покриттям Q , використовуючи термінологію T , якщо:

– ε є покриттям Q , використовуючи термінологію T , та

– не існує покриття ε' для Q , що використовує T , такого, що $(|Rest_\varepsilon Q|, |Miss_\varepsilon Q|) < (|Rest_{\varepsilon'} Q|, |Miss_{\varepsilon'} Q|)$, де $<$ – оператор лексикографічного порядку.

Найкраще покриття визначається як покриття, що, по-перше, має найменшу решту, та, по-друге, – найменшу частину відсутньої інформації. Задача найкращого покриття – це задача обчислення всіх найкращих покриттів Q , використовуючи термінологію T .

З технічної точки зору задача найкращого покриття належить до загальної структури для рерайтингу (перезапису), використовуючи термінологію з [9]. Ця структура визначається наступним чином: якщо задана термінологія T (тобто множина описів концептів), опис концепту Q , що не містить імен концептів, визначених в T , чи може Q може бути перезаписаний “кращим в деякому відношенні” описом ε , що побудований, використовуючи (деякі) імена, визначені в T ?

Формальне визначення задачі рерайтингу

Нехай N_R – множина імен ролей та N_P – множина примітивних імен, та нехай Λ_s, Λ_d та Λ_t – три ДЛ (відповідно, джерело, мета та $TBox$ -ДЛ). Задачі рерайтингу задаються наступними елементами:

Λ_t – $TBox$ -а T , що містить лише імена ролей з N_R та примітивні імена з N_P ; множина імен, що визначаються, які зустрічаються в T , позначається N_D ;

Λ_s – опис концепта C , що використовує лише імена з N_R та N_P ;

Бінарним відношенням $\rho \subseteq \Lambda_s \times \Lambda_d$ між описами Λ_s – та Λ_d - концептів.

Λ_d – рерайтингом концепта C , що використовує термінологію T , є опис $\varepsilon \in \Lambda_d$ – концепта, що побудований із використанням імен з N_R та $N_P \cup N_D$, такий, що $C \rho \varepsilon$.

Існуючі екземпляри цієї загальної структури можуть виділятися відносно природи відношення ρ , критерій оптимальності, як і мови Λ_s, Λ_d та Λ_t , відповідно, використовуються для опису в термінології T концепта (запита) Q та рерайтингу E . Прикладами таких екземплярів є: задача мінімально рерайтингу [9], де ρ встановлюється як еквівалентність по модулю T , тоді як розмір рерайтингу використовується в якості критерію оптимальності, та рерайтинг запитів, що використовує $views$ [10, 14], коли ρ встановлюється як включення та критерієм оптимальності є зворотне включення [9].

Відношення включення грає центральну роль в існуючих підходах рерайтингу. Дійсно, всі перелічені екземпляри загальної структури мають за мету переформулювати заданий запит Q в опис, що є еквівалентним або включається в Q .

Формально визначити задачу мінімального рерайтингу можна наступним чином.

Якщо задано відповідне відношення порядку \leq на Λ_d -концептах, рерайтинг E називається \leq -мінімальним, тоді і лише тоді, якщо не існує рерайтингу E' такого, що $E' < E$.

Задача мінімального рерайтингу – це екземпляр структури, де:

Всі три ДЛ – одна й та сама мова Λ ;

Бінарне відношення \leq відповідає еквівалентності $TBox$;

Описи Λ -концептів – впорядковані за розміром, тобто, $E < E'$, тоді й лише тоді, якщо $|E| < |E'|$. Розмір $|E|$ опису концепта E визначається як кількість входжень імен

концептів та ролей в E (де τ та \perp не враховуються).

Інтуїтивно, заданий рерайтинг повинен охопити всю 'інформацію', що надається запитом Q . Однак, у багатьох прикладних контекстах (наприклад, [15,16]) не реально вважати, що такий рерайтинг завжди існує, тому може бути доцільним розглядати рерайтинги, які апроксимують заданий запит. Це породжує новий екземпляр загальної структури для рерайтингу, а саме покриття концептів, що використовує термінології. Фактично, це і складає задачу найкращого покриття, де ρ відповідає поняттю покриття (тобто, це не еквівалентність, не включення), та критерієм оптимальності є мінімізація залишку (Rest) та нестачі (Miss) інформації.

Фактично, це і визначає задачу найкращого покриття, що сформульована вище. В даному випадку для рерайтингу використовується семантична різниця між концептами замість включення або еквівалентності, що робить процес рерайтингу більш гнучким.

Список використаних джерел

1. Introduction to XML [Електронний ресурс]. – Режим доступу: http://www.w3schools.com/xml/xml_whatIs.asp.
2. D. Fensel, C. Bussler, Y. Ding, and B. Omelayenko. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, 1(2), 2002.
3. D. Fensel, C. Bussler, and A. Maedche. Semantic Web Enabled Web Services. In *International Semantic Web Conference*, Sardinia, Italy, pages 1–2, Jun. 2002.
4. S. McIlraith, T.C. Son, and H. Zeng. Semantic Web Services. *IEEE Intelligent Systems. Special Issue on the Semantic Web*, 16(2):46–53, March/April 2001.
5. Srividya Kona, Ajay Bansal, Gopal Gupta Department of Computer Science The University of Texas at Dallas Richardson, TX 75083, Thomas D. Hite Metalect Corp. 2400 Dallas Parkway Plano, TX 75093 Automatic Composition of Semantic Web Services.

Відомості про автора:



Захарова Ольга Вікторівна – кандидат технічних наук, старший науковий співробітник Інституту програмних систем НАН України. Наукові інтереси: технології Semantic Web, онтології.

E-mail: ozakharova68@gmail.com

6. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and editors P. Patel-Schneider. *The Description Logic Handbook. Theory, Implementation and Applications*. Cambridge University Press, 2003.

7. F. Baader, R. Kuipers, and R. Molitor. Computing Least Common Subsumer in Description Logics with Existential Restrictions. In T. Dean, editor, *Proc. of the 16th Int. Joint Conf. on AI*, pages 96–101. M.K., 1999.

8. G. Teege. Making the difference: A subtraction operation for description logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *KR'94*, San Francisco, CA, 1994. Morgan Kaufmann.

9. Franz Baader, Ralf Kuipers, and Ralf Molitor. Rewriting Concepts Using Terminologies. In *Proc. of the Int. Conf. KRColorado, USA*, pages 297–308, Apr. 2000.

10. C. Beeri, A.Y. Levy, and M-C. Rousset. Rewriting Queries Using Views in Description Logics. In L. Yuan, editor, *Proc. of the ACM PODS*, New York, USA, pages 99–108, Apr. 1997.

11. F. Goasdoué and M-C Rousset V. Lattès. The Use of CARIN Language and Algorithms for Information Integration: The PICSEL System. *IJICIS*, 9(4):383–401, 2000.

12. Alon Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.

13. Franz Baader, Ralf Kuipers, and Ralf Molitor. LuFg Theoretische Informatik, RWTH Aachen. Computing Least Common Subsumers in Description Logics with Existential Restriction, 1999.

14. Alon Y. Halevy. Answering queries using views: A survey. *VLDB Journal*, 10(4):270–294, 2001.

15. B. Benatallah, M-S. Hacid, A. Leger, C. Rey, and F. Toumani. On automating Web services discovery. *VLDB J.*, 14(1):84–96, 2005.

16. B. Benatallah, M-S. Hacid, H-Y. Paik, C. Rey, and F. Toumani. Towards semantic-driven, flexible and scalable framework for peering and querying e-catalog communities. *Information Systems*, 31(4-5), 2006.