УДК 004.94

О.В. Захарова

СЕМАНТИЧНА АНОТАЦІЯ ВЕБ-СЕРВІСІВ

Важливість опису Веб-сервісів на рівні процесів та створення механізмів для роботи з такими описами широко визнана. Визначення та розуміння клієнтом семантик кожного елементу такого процесу забезпечує можливість їх адекватного вибору та застосування. Використання та формалізація семантичних анотацій відкриває шлях до автоматизованого вирішення такої складної задачі Веб-сервісів, як їх композиція. В даній роботі визначається роль анотацій. Типи семантик, що надаються в анотаціях, розглядаються у взаємозв'язку з існуючими задачами Веб-сервісів на всіх етапах їх життєвого циклу. При цьому Веб-сервіс розглядається як з точки зору даних, так і поведінки. Наведені основні етапи анотування. Розглядається проблема мінімізації анотацій, як механізму підвищення ефективності вирішення проблем Веб-сервісів. Визначається місце та роль дескриптивних логік в семантизації Веб-сервісів.

Ключові слова: семантичний Веб-сервіс, дескриптивна логіка, типи семантик, семантична анотація, процесна модель, композиція семантичних Веб-сервісів, етапи анотування, файл анотації, мінімізація анотацій.

Вступ

Останнім часом сервіс-орієнтована архітектура (SOA) набула стану широко використовуваної парадигми, де основою для вирішення бізнес-задачі є Веб-сервіси. З розвитком цієї галузі підприємства переходять від використання стандартів Веб-сервісів для встановлення статичних зв'язків між компонентами до необхідності дослідження динамічних пропозицій, таких як повторне використання сервісів, їх взаємодія, автоматична побудова складного композитного сервісу, що буде реалізовувати конкретну бізнес-задачу. Блоки побудови рішень, на базі SOA, це окремі, самостійно-описані Веб-сервіси, що можуть повторно використовуватися різними прикладними системами.

Щоб правильно обрати та використовувати Веб-сервіс провайдера, клієнт повинен розуміти семантики кожної його операції. Іншими словами, мати можливість однозначно розшифрувати ціль, на яку спрямована кожна з операцій, як і передбачуваний зміст усіх її параметрів. Для цього необхідно:

- узагальнена модель домену або онтологія предметної області (ПО), де погоджені всі терміни (імена операцій та параметрів) між провайдерами сервісів та клієнтами;
- коментування всіх аспектів сервісу;
- елементи сервісу мають анотуватися термінами з моделей домена (онтології ПО).

Зрозуміло, що анотування Веб-сервісу засобами формальних мов моделювання забезпечить можливість машинної обробки анотацій та прийняття рішення щодо використання сервісу. З іншого боку, надмірна семантична інформація суттєво перевантажить опис Веб-сервісу та зробить складнішим його використання. Тому, головна мета цього дослідження полягає у тому, щоб на базі аналізу існуючих підходів:

- визначити роль та місце семантик у процесі пошуку ефективного вирішення проблеми автоматизованої композиції семантичних Веб-сервісів, що представлені процедурним описом;
- визначити принципи та важелі оптимальності використання семантичних описів з метою підвищення ефективності вирішення задач Веб-сервісів;
- визначити місце та доцільність використання апарату ДЛ для визначення семантик сервісів;
- дійти висновків щодо ефективних методів формального визначення самих семантик.

Задачі Веб-сервісів та типи семантик

Веб-сервіс можна розглядати з точки зору його даних та поведінки. Опис даних це фактично визначення типів даних, що використовуються сервісом. Поведінковий аспект – це опис операцій сервісу. Відповідно, семантично анотування Веб-сервісів передбачає пояснення точних семантик елементів даних та функціональності Веб-сервіса. Це має вирішальне значення для його використання та здійснюється шляхом анотування елементів Веб-сервісів концептами моделей домену або онтологій. Онтології представляють погоджений погляд на змодельований домен сервісів, тому виключається будь-яка неоднозначність в інтерпретації функціональності або даних Веб-сервісу. Мета анотування Веб-сервісів [1] полягає у тому, щоб створити можливість недвозначного та автоматизованого виявлення та композиції сервісів.

Щоб визначити, яку частину Веб-сервіса потрібно анотувати, важливо розуміти взаємодію семантик в життєвому циклі сервіса та як ці семантики використовуються у Веб-сервісі.

При вирішенні задач виявлення Веб-сервісів або синтезу складного композитного Веб-сервісу, запитувач описує свої вимоги до потрібного сервісу в термінах його функціональності (операцій Веб-сервісів) та даних, що ним використовуються, а саме – входів та виходів (наприклад, які відношення існують чи повинні існувати між даними, поданими на вхід сервісу, і даними, отриманими як відповіді від сервісу).

Окрім цього, специфікації сервісів можуть включати передумови та ефекти. Передумови – це вимоги, які мають задовольнятися до виконання операції Веб-сервісу, а ефекти є результатами виконання операції. Так, наприклад, анотації можуть дозволити описати ефекти, асоційовані з успішними виконаннями, а це, в свою чергу, забезпечить ідентифікацію успішних виконанньь сервісу і дозволить відрізнити їх від збоїв. Таким чином, семантичні анотації пов'язані з входами, виходами, передумовами та ефектами елемента операції Веб-сервіса.

Більш розвинені механізми виявлення розглядають також не функціональні аспекти Веб-сервісів [2, 3] та вимоги споживача, такі як метрики якості, надійність, безпека, вартість, ступінь довіри і т. і.

Вирішуючи задачі на рівні процесів, потрібна можливість виражати вимоги на поведінку сервісу. Цього можна досягти за допомогою використання мов представлення, що засновані на темпоральній логіці (наприклад, CTL [4]), збагачених твердженнями концепту та ролі дескриптивної логіки (ДЛ).

Додавання семантик доцільне та має свої переваги на всіх етапах життєвого циклу Веб-процеса. Розробники можуть використовувати семантичні анотації для пояснення можливостей своїх Веб-сервісів.

Запитувач може формулювати свої вимоги для виявлення або композування Веб-сервісів у шаблоні семантичного сервісу [5]. Шаблон семантичного сервісу – це абстрактний опис сервісу, де потік управління створюється вручну, а потрібна функціональність описується, використовуючи терміни з моделі домена або онтології.

Техніки міркувань можуть використовуватись для порівняння вимог в шаблоні сервісу з можливостями Веб-сервісів, що доступні в реєстрі (UDDI), для виявлення сервісів. В ході композиції, функціональний аспект анотацій може використовуватися для створення корисних композицій сервісів.

Виходячи із загальних задач та проблем Веб-сервісів, можна виділити наступні типи семантик (таблиця) [6].

Якісне вирішення комплексної задачі побудови композитного сервісу, що відповідатиме бізнес-цілі, потребує використання всіх наведених типів семантик.

При вирішенні вище наведених задач Веб-сервіси можна класифікувати щодо моделей їх взаємодії:

- атомні Веб-сервіси, що розглядаються як «чорна скринька», тобто описуються своїми параметрами (вхід, вихід, передумови, післяумови);

Таблиця

Тип семантик	Опис	Задачі, де використовується
Семантики даних	Формальне визначення даних у вхідних та вихідних повідомленнях Веб-сервісів	Задачі виявлення сервісів та забезпечення інтероперабельності між сервісами
Функціональні семантики	Формальне визначення функціональності сервісу	Задачі виявлення та композиції Веб-сервісів
Нефункціональні семантики	Формальне визначення кількісних або некількісних обмежень таких як: якість сервісу, мінімальна вартість та правила доступу, вимоги до кодування повідомлень	Задачі виявлення, композиції та інтероперабельності Веб-сервісів
Семантики виконання	Формальний опис виконання або потоку сервісів в процесі або операціях всередині сервісу	Процес верифікації або обробки виключень. Верифікація включає перевірку коректності (поток даних та управління) композиції сервісів. Мета обробки виключень полягає у ідентифікації точок розриву у Веб-процесі та визначенні, як це подолати та діяти далі від таких точок

• поведінкові, Веб-сервіси, що базуються на поведінковій (процесній) моделі, часто називаються «сірою скринькою», описуються порядком виконання операцій.

Нас будуть цікавити, перш за все, проблеми анотування сервісів, що представлені поведінковою моделлю. Поведінкові моделі сервісу можуть використовуватися для виконання завдань перевірки, виявлення, вибору і композиції, та для вираження вимоги до сервісу на рівні процесу [7]. Всі ці завдання потребують перевірки певних умов на поведінку і семантику даного сервісу. Наприклад, якщо розглянути задачу вибору процесу, то ми маємо набір процесів і специфікацію користувача, вибір полягає у розпізнаванні, які процеси задовольняють дану специфікацію. Використання семантики дозволяє виразити і процеси, і специфікації як семантично збагачені моделі.

У випадку поведінкового представлення інтерфейс сервісу описує процес, що взаємодіє з іншими сервісами на різних стадіях, які можуть мати різні керуючі конструкції, наприклад, послідовність, умова і ітерація. Таке представлення включає поняття стану, і такий Веб-сервіс може бути відображений у систему перехідного стану (STS) [8].

Етапи семантизації сервісу

Кінцевою метою вирішення більшості задач Веб-сервісів є ефективне вирішення бізнес-задач, шляхом синтезу відповідного Веб-сервісу, що її реалізує. Так як, нас цікавить поведінкова модель сервісу, ми розглядаємо сервіси, що описані як системи з фіксацією станів. Тобто мова йде про створення методів автоматизованої композиції семантичних Веб-сервісів, представлених процедурним описом. Семантичні описи таких Веб-сервісів відкривають шлях до автоматизації їх композиції. Але, слід зазначити, що методи автоматизованої композиції повинні використовувати обмежену, корисну і задовільну, кількість семантичних міркувань.

Одна з ключових ідей досягнення цієї мети полягає в окремому збереженні процедурного опису процесів і онтологічних описів, і додавання семантичних анотацій, які їх пов'язують.

Процедурна поведінка Веб-сервісу описується мовами, які були розроблені для опису процесів, а семантика даних, якими обмінюються процеси, – в окремій онтологічній мові. Та нарешті, ці два описи повинні бути пов'язані семантичними анотаціями поведінкових описів, які зіставляються з онтологічними концептами.

В роботі [1] пропонується підхід, де пропонується алгоритм заземлення (grounding), що включає семантичні анотації в STS, що моделює Веб-сервіси. Це дозволяє отримати STS модель, яка опрацьовується існуючими контролерами моделей (model checker) [9] та планувальниками [10], щоб вирішити проблеми верифікації, вибору та композиції. Запропонований підхід спрямований головним чином на збереження оригінального синтаксису BPEL [11] і WSDL [12] файлів, і анотація поміщається в інший файл з посиланнями на BPEL і WSDL через вирази XPath.

Дана методологія [1] містить чотири етапи: анотація (annotation), конвертування моделі (model translation), заземлення (grounding), перевірка моделі (model checking).

Анотація. Визначення даних і поведінки процесу збагачуються посиланнями на онтологію. Онтологія повинна бути загальноприйнятою формалізацією деякої області.

Конвертування моделі. Вираження процесу в іншій формі, яка може бути перевірена легко і автоматично. Зокрема, оскільки контролери моделі (model checker) зазвичай мають справу з деяким видом систем перехідних станів, доцільно переводити анотований BPEL процес в анотовану систему переходів станів (Annotated State Transition Systems - ASTS). Цей крок може виконуватися автоматично. Відомі також розробки по конвертуванню OWL-S процесів в HTN-DL [13], що дозволяє одразу застосовувати дану систему AI планування для вирішення задачі автоматичної композиції сервісів. Але алгоритми транслювання BPEL-процесів HTN-DL та питання доцільності такого перетворення є предметом подальших досліджень.

Заземлення (grounding) – це процедура, за допомогою якої семантичні анотації "звужуються" до чисто синтаксичної форми, грубо кажучи, твердження концептів і ролі перетворюються в логічні висловлювання. З технічної точки зору, кожен анотований стан – це база знань. Тому, дізнатися, які твердження виконуються в цьому стані, можна за допомогою query answering service [14] ДЛ. Це гарантує, що анотації можна трактувати за допомогою існуючих контролерів моделі, які працюють тільки з висловлюваннями (proposition). Заземлення повинно застосовуватися як до формального представлення процесу, так і до специфікації мети. У випадку, якщо ми конвертували анотований BPEL процес в анотовану STS, заземлення застосовується до анотованої STS і до специфікації мети, яка може бути виражена в анотованій CTL. Результатом виконання алгоритму заземлення, у такому випадку, є заземлена (пропозиціональна) STS і заземлена (пропозиціональна) CTL специфікація.

Фаза перевірки моделі полягає у перевірці, чи підтверджується специфікація заземленої мети (наприклад, пропозиціональна CTL) заземленою моделлю процесу (наприклад, пропозиціональна STS).

Використання ДЛ для анотування Веб-процесів

Стратегія анотації Веб-сервісів семантикою повинна охоплювати як статичні анотації, що «орієнтовані на дані», так і процедурні – «орієнтовані на процес».

Відповідно до понять дескриптивної логіки [15], в межах даної, загальноприйнятої онтології існує термінологічний компонент (T-BOX) [16] і стверджувальний (assertional) компонент (A-BOX). T-BOX містить визначення концептів, а також відношення узагальнення та агрегації між ними. Вважаємо, що T-BOX є єдиним для всіх сервісів, які необхідно анотувати в домені. Він містить всі концепти, які необхідно представляти в домені застосування. Якщо анотації винести до окремого файлу (із посиланнями на файл процесу), то це дозволить глобальні твердження, які дійсні для всіх станів процесу, а саме твердження T-BOX, не повторювати в описі кожного

стану процесу, а визначити однократно у відповідному розділі файлу анотацій.

A-BOX містить визначення твердження двох різних типів: твердження концептів та твердження ролі. З кожним станом кожного сервісу пов'язується свій A-BOX. Він описує наслідки даної дії в термінах тверджень концепту і ролі. Тим не менш, є деякі твердження, які не залежать від будь-яких дій, але виконуються скрізь. Такі твердження завжди істині, незалежно від того, як розвивається процес.

Твердження концептів мають вигляд: *a : C*. Такий вираз означає, що індивід *a* є екземпляром концепту *C*. Твердження ролі визначають значення, які мають певні ролі індивідуумів. Вони мають форму *a.R = b*. Інтуїтивно, такий вираз означає, що значення атрибуту *R* індивіда *a* є *b*, де *b* є інший індивід або літерал. Ролі, які використовуються в твердженні ролі, визначаються в термінологічній частині онтології.

Розглянемо як приклад задачу забезпечення сервісу віртуального агенства [17] шляхом композиції існуючих транспортних і готельних сервісів. Призначення сервісу – визначити прийнятний пакет відпустки, відповідно до запитів користувача. Вибір постачальників сервісів може залежати від обмежень, які задані кінцевим користувачем і знаннями в предметній області. Так, наприклад, він може враховувати особливості подорожі залежно від міста призначення та тривалості, допустимих типів розташування та транспортних засобів. Припустимо, що замовник хоче побудувати такий сервіс для короткострокових подорожей, та накладає наступні обмеження: подорож може здійснюватися літаком або потягом з зупинкою в місті призначення в готелі або гостьовому будинку. Враховуючи наведені обмеження, можна дійти висновку, що для побудови результуючого сервісу нам необхідно композувати сервіси:

- бронювання квитків на літак,
- бронювання квитків на потяг,
- бронювання готелів, та
- бронювання гостьових будинків.

Подібні за функціональністю сервіси різних провайдерів можуть містити різний ланцюжок операцій та використовувати різні онтології домену. Тому, необхідно, перш за все, створити єдину узагальнену онтологію. В даному випадку відповідний узагальнений TBox може містити наступні твердження:

$$
\begin{aligned}
\text{Date} &= \forall \text{year.Number} \sqcap \forall \text{month.Number} \sqcap \forall \text{day.Number} \\
\text{Client} &= \forall \text{name.String} \sqcap \forall \text{gender.Gender} \\
\text{Gender} &\sqsubseteq \top \\
\text{Status} &\sqsubseteq \top \\
\text{Location} &= \forall \text{name.String} \\
\text{Trip} &= \forall \text{id.String} \sqcap (\leq 1\text{id}) \sqcap (\geq 1\text{id}) \sqcap \forall \text{date.Date} \sqcap \\
&\quad \forall \text{start.Location} \sqcap \forall \text{destination.Location} \sqcap \forall \text{pax.Client} \sqcap \\
&\quad \forall \text{status.Status} \\
\text{Accomodation} &= \forall \text{id.String} \sqcap (\leq 1\text{id}) \sqcap (\geq 1\text{id}) \sqcap \forall \text{date.Date} \sqcap \\
&\quad \forall \text{location.Location} \sqcap \forall \text{pax.Client} \sqcap \\
&\quad \forall \text{status.Status} \\
\\
\text{Flight} &= \text{Trip} \sqcap (\leq 1\text{id}) \sqcap (\geq 1\text{id}) \sqcap \\
&\quad \forall \text{seatNumber.String} \\
\text{Train} &= \text{Trip} \sqcap (\leq 1\text{id}) \sqcap (\geq 1\text{id}) \sqcap \\
&\quad \forall \text{seatNumber.String} \\
\text{Hotel} &\sqsubseteq \text{Accomodation} \sqcap \forall \text{roomNumber.String} \\
\text{GuestHouse} &\sqsubseteq \text{Accomodation} \sqcap \forall \text{roomNumber.String}
\end{aligned}
$$

В наведеному описі термінології сутності *Date, Client, Location, Trip, Accommodation* складають спільну частину домену, та їх можна розглядати як частину знань домену. Решта понять (*Flight,Train, Hotel, GuestHouse*) є специфічними для Веб-сервісів, які використовуються в композиції і можуть бути отримані шляхом відображення локальної онтології кожного Веб-сервісу в загальну онтологію.

Визначення понять *Trip* та *Accommodation* містять статус ролі, значеннями яких є концепт *Status*. Ця роль фіксує поточний стан запиту клієнта и може приймати наступні значення: *Available* – якщо поїздка (проживання) доступні; *NotAvailable* – коли поїздка (проживання) не доступні, *booked* – коли поїздка (проживання) замовлені та *cancelled* коли поїздка (проживання) анульовані. Ці значення фактично є екземплярами концепту *Status* і повинні бути визначені в ABox:

available : Status, notAvailable : Status, booked : Status, cancelled : Status

Окрім цього, концепт *Gender* може приймати лише одне з двох значень *male*

або *female*, і вони теж повинні бути перераховані в АВох:

male : Gender, female : Gender

ДЛ є формальною мовою, підтримує концепцію відкритого світу та власні механізми міркувань. Все це робить її бажаною та ефективною як для представлення функціональної частини опису Веб-сервісу, так і для представлення семантичних елементів (анотацій) у процедурному описі Веб-сервісу. А така властивість ДЛ, як підтримка можливостей логічного виведення, забезпечує ефективність вирішення багатьох задач Веб-сервісів, як елементів загальної складної задачі композиції.

Розширення процедурного опису сервісу семантичними анотаціями

Як вже було зазначено вище, для процедурного опису сервісу використовується абстрактний BPEL. BPEL [11] забезпечує операційний опис поведінки Веб-сервісів на верхній частині інтерфейсів сервісів, які визначені в специфікації WSDL. Абстрактний опис BPEL визначає партнерів сервісу, його внутрішні змінні і операції, які спрацьовують від виклику сервісу деякими з партнерів. Операції включають в себе призначення змінних, виклик інших сервісів та отримання відповідей, породження паралельних потоків виконання і недетермінованого вибору одного серед різних напрямів дій.

Так, наприклад, для сервісу резервації польотів абстрактний BPEL опис може мати наступний вигляд (опис та склад операцій схожих за функціональністю сервісів може відрізнятися для сервісів різних провайдерів) [17]:

```
<process name="FlightReservation">
    <partnerLinks>
        <partnerLink name="client"
    partnerLinkType="FtRes_PLT"
    myRole="FtRes_Server"
    partnerRole="FtRes_Client"/>
    </partnerLinks>
    <variables>
        <variable name="req"
    messageType="flightRequest"/>
            <!-- "req" contains parts
        "/req/start", "/req/des", and
        "/req/date" --->
        <variable name="pax"
messageType="paxInformation"/>
            <!-- "pax" contains part
        "/offer/client" -->
        <variable name="offer"
messageType="flightOffer"/>
            <!-- "offer" part "/offer/fl" --
        >
</variables>
<sequence name="main">
        <receive operation="request"
variable="req" partnerLink="client"
          semann="/req/start :
Location, /req/dest : Location, /req/date
: Date"/>
        <switch
name="checkAvailability">
          <case name="isNotAvailable">
            <invoke
        operation="not_avail"
        partnerLink="client"
        semann="/offer/fl : Flight,
        /offer/fl.status = notAvailable"/>
          </case>
          <otherwise
name="isAvailable">
            <assign
        name="prepareOffer">
            <copy><from
        opaque="yes" semann="/offer/fl
        : Flight, /offer/fl.start =
        /req/start,
                    /offer/fl.destinati
            on = /req/dest,
            /offer/fl.date =
            /req/date"/>
                <to
            variable="offer"
            part="fl"/></copy>
            </assign>
            <invoke operation="offer"
        inputVariable="offer"
        partnerLink="client"/>
            <pick
        name="waitAcknowledge">
                <onMessage
            operation="ack"
            cariable="pax"
            partnerLink="client"
                semann="/pax/cl
            ient : Client, /offer/fl.pax
            = /client/pax,
            /offer/fl.status =
            booked"/>
                <onMessage
            operation="nack"
            partnerLink="client"
```

```
                    semann="/offer/flight.st
                    atus = cancelled"/>
                    </pick>
                </otherwise>
        </switch>
        </sequence>
</process>
```

BPEL специфікація досить детально описує взаємодії, які необхідно виконати з Веб-сервісом для його використання. Але вона не містить опису семантичних аспектів таких взаємодій, і тому не є достатньою для автоматичної композиції такого Веб-сервісу з іншими. Тому, BPEL специфікації розширюються «семантичними анотаціями». У наведеному вище описі такі семантичні анотації є значеннями атрибутів *semann*.

Специфікацію можна умовно розділити на дві частини: декларація змінних, які використовуються у вхідних/вихідних повідомленнях, та решта частина абстрактної специфікації BPEL, що описує потік взаємодії.

Наведений приклад досить наочно демонструє основні аспекти використання семантичних анотацій, а саме:

- для забезпечення зв'язку понять в онтології з вхідними та вихідними повідомленнями, якими обмінюється процес.

В наведеному прикладі цю роль відіграють семантичні анотації *«/req/start : Location, /req/dest : Location, /req/date : Date»* діяльності receive для операції request на початку опису;

- щоб виразити «семантичні» відношення між значеннями вхідних і вихідних параметрів, якими обмінюються Веб-сервіси при взаємодії. В наведеному прикладі, це визначається в анотації *«/offer/fl.start = /req/start, /offer/fl.destination = /req/dest, /offer/fl.date = /req/date»;*

- для визначення результату взаємодії з Веб-сервісом. У даному прикладі, рейс бронюється лише за умови його доступності. Сервіс відсилає пропозицію і користувач визнає згоду з пропозицією. Щоб виразити це в BPEL специфікації, в діяльність, що відповідає за отримання підтвердження, додано анотацію *«/offer/status = booked»*.

Таким чином, семантичні анотації необхідні, щоб корегувати особливості інтерфейсу, і визначати це у відношенні із загальною онтологією. Але включення семантичних анотацій до процедурного опису значно його перевантажують, збільшують розмір BPEL-файлу. Тому, слід зазначити доцільність:

- використання лише необхідних анотацій, що безпосередньо впливають на ефективність вирішення цільової задачі, а саме лише визначених вище типів семантик;

- зберігання анотацій в окремому файлі (окремо від процедурного опису).

Схема файлу анотації

Як вже було зазначено, утримання анотації окремо від процедурного опису сервісу значно спрощує опис процесу, перш за все, завдяки тому, що загальна інформація, яка істинна у всіх його станах, в описі кожного стану не повторюється. В такому випадку, загальний опис Веб-сервісу, що надає інформацію про дані, які використовуються сервісом та представленим інтерфейсом, зберігаються в окремому файлі. Таким чином, ми маємо процедурний опис Веб-сервісу, узагальнену онтологію домену та файл анотації в окремому файлі. Це може бути файл, визначений на синтаксисі XML, який буде зв'язуватися з відповідними файлами процедурного опису та онтології за допомогою XPath запитів.

Виходячи з визначених вище типів семантик та місця і ролі анотацій у вирішенні задач Веб-сервісів, файл анотації повинен містити наступні розділи (XML схема [1]).

Анотації типів даних. Вводять обмеження на процедурні анотації, тобто A-BOX-и, пов'язані з системою перехідного стану.

Декларації. У цьому розділі перераховуються всі індивіди, які використовуються в твердженнях.

Глобальні твердження. Являють собою твердження концепту та ролі, які

виконуються у кожному стані процесу. Вони є синтаксичним цукром (sintactic sugar), так як вони визначаються тільки, щоб уникнути їх повторення в кожному стані.

Процедурна анотація містить набори тверджень концептів та ролі та завжди звертається до конкретної операції процесу (BPEL дії). Інтуїтивно, процедурна анотація містить всі твердження, які виконуються після того, як відповідна дія була виконана.

Висновки

Кінцевою метою, що лежить в основі виконання більшості задач Веб-сервісів, є ефективна побудова складного Веб-сервісу, що реалізує конкретну бізнес-задачу. Композиція Веб-сервісів має дві цілі: з одного боку – задовільнити складні вимоги клієнта, а з іншого – зменшити складність розробки Веб-сервісу, щоб задовільнити складну прикладну систему.

До вирішення цієї задачі існують синтаксичні та семантичні підходи. До прикладів семантичних підходів та технік до вирішення задачі композиції можна віднести техніки, що базуються на AI-плануванні, підходи, що базуються на алгоритмах ланцюжка, техніки композиції на основі інтерфейсу автоматів. Щоб семантично описати Веб-сервіс використовуються анотації. Сервіс анотується семантичною моделлю, яка забезпечує семантичну інформацію щодо його функціональності (входи, виходи). Процес анотування полягає у виділенні функціональних компонент та встановлення зв'язку кожної функціонального компонента з відповідним елементом семантичної моделі.

В даній роботі визначено роль та місце семантичних анотацій у процесі пошуку ефективного вирішення проблеми автоматизованої композиції семантичних Веб-сервісів, представлених процедурним описом. Сформульовані основні задачі Веб-сервісів на процесному рівні та визначені відповідні типи семантик, які вони використовують. Описані етапи анотування та визначене місце дескриптивних логік у створенні анотованих описів Веб-сервісів.

Напрямком подальших досліджень є визначення ефективності використання семантичних анотацій та їх місця з точки зору використання технік AI-планування для синтезу композитного семантичного Веб-сервісу, аналіз можливості та доцільності застосування існуючих технік AI-планування, та проблем і розбіжностей, які при цьому можуть виникнути.

1. *Kunal Verma.* Semantically Annotating a Web Service / Accenture Technology Labs Amit Sheth/Wright State University. – www.computer.org/internet
2. *Holger Lausen, Francisco Martin-Recuerda, Jos de-Bruijn, and Michael Stollberg.* A Conceptual and Formal Framework for Semantic Web Services (v1) / University of Innsbruck. – http://www.sti-innsbruck.at/results/publications/deliverables/conceptual-and-formal-framework-semantic-web-services-v1
3. *Sudhir Agarwal.* Formal Description of Web Services for Expressive Matchmaking. / Dipl.-Inform., Karlsruhe, 2007.
4. *Emerson E.A.* Temporal and modal logic. / In J. van Leeuwen, editor, Handbook of Theoretical Computer Science, volume B: Formal Models and Semantics // Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, New York, N.Y., 1990. – P. 996–1072.
5. *Sivashanmugam K., Verma K., Sheth A., Miller J.* Adding Semantics to Web Services Standards // Proceedings of the 1st International Conference on Web Services (ICWS'03), Las Vegas, Nevada, June 2003. – P. 395–401.
6. *The* Web Service Modeling Framework. – http://www.wsmo.org/.
7. *Sudhir Agarwal*. A goal specification language for automated discovery and composition of web services./ In International Conference on Web Intelligence (WI '07), Silicon Valley, USA, NOV 2007.
8. *Pistore M., Traverso P., Bertoli P., and Marconi A.* Automated Synthesis of Composite BPEL4WS Web Services // In Proc. ICWS'05, 2005.
9. *Cimatti A., Clarke E.M., Giunchiglia F. and Roveri M.* NUSMV: a new symbolic model checker // International Journal on Software Tools for Technology Transfer, 2(4). – 2000.

10. *Bertoli P., Cimatti A., Pistore M., Roveri M. and Traverso P.* MBP: a Model Based Planner // In IJCAI-2001 workshop on Planning under Uncertainty and Incomplete Information, 2001.
11. *Andrews T., Curbera F., Dolakia H. and al.* Business Process Execution Language for Web Services (version 1.1), 2003.
12. *WSDL* – http://www.w3.org/TR/wsdl
13. *Evren Sirin.* Combining Description Logic Reasoning with AI Planning for Composition of WEB Services./ Dissertation submitted to the Faculty of the Graduate School of the University of Maryland, College Park in partial fulfillment of the requirements for the degree of Doctor of Philosophy, 2006.
14. *Baader F. and Nutt W.* Basic Description Logics. *In Baader F., Calvanese D., McGuinness D., Nardi D., and Patel-Schneider P. editors.* The Description Logic Handbook: Theory, Implementations and Applications, Cambridge University Press, 2003. – P. 43–95.
15. *Ortiz M., Calvanese D., and Eiter T.* Characterizing Data Complexity for Conjunctive Query Answering in Expressive Description Logics // AAAI. – 2006.
16. *Staab S., Studer R.* Handbook on Ontologies., Second edition, 2009.
17. *Marco Pistore, Luca Spalazzi, and Paolo Traverso.* A Minimalist Approach to Semantic Annotations for Web Processes Compositions // Universit`a di Trento – Via Sommarive 14 – 38050 Povo – Trento – ITALY pistore@dit.unitn.it, Universit`a Politecnica delle Marche – Via Brecce Bianche – 60131 Ancona – ITALY spalazzi@diiga.univpm.it, ITC-irst – Via Sommarive 18 – 38050 Povo – Trento – ITALY traverso@irst.itc.it, 2004.

Одержано 09.09.2015

Про автора:

Захарова Ольга Вікторівна;
кандидат технічних наук,
старший науковий співробітник.
Кількість наукових публікацій в
українських виданнях – 24.
ORCID orcid.org/0000-0002-9579-2973.

Місце роботи автора:

Інститут програмних систем
НАН України,
Проспект Академіка Глушкова, 40.
Тел.: 526 5139.
E-mail: ozakharova68@gmail.com