УДК 004.94

СПЕЦИФІКАЦІЯ ПРОЦЕСУ СЕМАНТИЧНОЇ АНОТАЦІЇ ВЕБ-СЕРВІСІВ

О.В. Захарова

Складність бізнес-задач, для вирішення яких використовуються веб-сервіси, не дозволяє обмежити їх розгляд функціональною моделлю, а вимагає розгляду й поведінкових аспектів. А забезпечення можливостей автоматизованого вирішення задач веб-сервісів вимагає, перш за все, семантизації їх опису з використанням формальних засобів представлення, що будуть зрозумілими для комп'ютера. Сервіси, що не збагачені семантикою, не надають можливості міркування про те, що робить цей сервіс. Семантизація полягає у розширенні опису веб-сервісу чіткими і зрозумілими семантичними анотаціями, що повинні охоплювати всі важливі функціональні та не функціональні аспекти сервісу, але не перевантажувати його надмірною інформацією. Тому, досить актуальними є проблеми специфікації самого процесу анотування, враховуючи визначення мети семантизації, основних аспектів, що підлягають анотуванню, джерел інформації, категорій семантик та кроків процесу, який забезпечує отримання анотованої специфікації сервісу, з подальшим перетворенням у форму, яка може оброблятися автоматично, що і є предметом розгляду даної роботи. Основні етапи процесу відповідають категоріям семантик, що визначаються в процесі. Для кожного етапу специфікуються входи, виходи, джерела інформації та кроки реалізації. Окрім цього, в роботі особлива увага приділяється визначенню місця дескриптивної логіки у процесі семантизації веб-сервісу, як формального інструменту семантичного опису сервісу, що забезпечує можливості перевірки коректності опису та логічного виведення при вирішенні задач веб-сервісів тощо. Наводиться приклад розширення онтологічними семантичними анотаціями фрагменту BPEL-процесу. В даному випадку семантизація полягає у виборі відповідних онтологій доменів та встановленні зв'язків між концептами онтології, яка представлена за допомогою апарату дескриптивних логік, та метаданими, що описують функціональність веб-сервісу. Як зрозуміла для комп'ютера модель представлення сервісу обрано анотовану систему перехідних станів (ASTS). Це обумовлюється тим, що з STS, зокрема, працюють більшість систем AI планування, механізми яких можуть також використовуватися для автоматичного вирішення задач веб-сервісів.
Ключові слова: веб-сервіс, задачі веб-сервісів, семантизація веб-сервісу, категорії семантик, процесна модель, інтерфейс, передумова, постумова, ефект сервісу, семантична анотація, процес анотування.

Сложность решаемых с помощью веб-сервисов задач не позволяет ограничить их рассмотрение функциональной моделью, а требует рассмотрения также и поведенческих аспектов веб-сервисов. Для обеспечения возможности автоматизированного решения задач веб-сервисов, прежде всего, необходимо семантизировать их описание, используя понятные компьютеру формальные средства представления. Не обогащенные семантикой сервисы не позволяют судить о том, что делает этот сервис. Семантизация заключается в расширении описания веб-сервиса четкими и понятными семантическими аннотациями, которые должны охватывать все важные функциональные и не функциональные аспекты сервиса, но при этом не перегружать его избыточной информацией. Этим объясняется актуальность проблемы спецификации самого процесса аннотирования, с учетом определения цели семантизации, основных аспектов сервиса, подлегающих аннотированию, источников информации, категорий семантик и шагов процесса, обеспечивающего получение аннотированной спецификации сервиса, с дальнейшей конвертацией – преобразованием в машинно-обрабатываемую форму, что и является предметом рассмотрения данной работы. Основные этапы процесса соответствуют категориям семантик, определяемых при аннотировании. Для каждого этапа специфицируются входы, выходы, источники информации и шаги реализации. Кроме этого в работе особенное внимание уделяется определению места дескриптивной логики в процессе семантизации веб-сервиса, как формального инструмента семантического описания сервиса, обеспечивающего возможности проверки корректности описания и логического вывода при решения задач веб-сервисов и т.п. Приводится пример расширения фрагмента BPEL-процесса онтологическими семантическими аннотациями. В данном случае, семантизация состоит в выборе подходящих онтологий доменов и установлении связей между концептами онтологии, представленной средствами аппарата дескриптивных логик, и метаданными, описывающими функциональность веб-сервиса. В качестве понятной для компьютера модели представления сервиса выбрана аннотированная система переходных состояний (ASTS). Это обуславливается тем, что, в частности, с STS работает большинство систем AI планирования, механизмы которых могут тоже использоваться для автоматического решения задач веб-сервисов.
Ключевые слова: веб-сервис, задачи веб-сервисов, семантизация веб-сервиса, категории семантик, модель процесса, интерфейс, предусловие, постусловие, эффект сервиса, семантическая аннотация, процесс аннотирования.

The complexity of the tasks solved with web services does not allow to limit their consideration to a functional model, but also requires the consideration of behavioral aspects of web services. To provide the possibility of automated resolving web services tasks, first, it is necessary to semantify their description using computer-friendly tools for formal definition. Web service that is not enriched with semantics, does not allow you to understand what it does. Semantization consists in expanding the description of the web service with clear and understandable semantic annotations that should cover all important functional and non-functional aspects of the service, but do not overload it with redundant information. This explains the relevance of the problem of the specification of the annotation process itself, taking into account the definition of the semantization goal, the main aspects of annotating the service, the sources of the information, the semantics categories, and the process steps providing the annotated specification of the service, with further conversion - transformation into a machine-processed form. It is the subject of this paper. The main stages of the process correspond to the categories of semantics defined in the annotation. Inputs, outputs, information sources and implementation steps are specified for each stage. Besides special attention is paid to define the place of descriptive logic in the process of semantizing the web service. Descriptive logic is considered as a formal tool for the semantic description of the service, which makes it possible to verify the correctness of the service's definition and it allows logical inference when solving Web services tasks, etc. It is given an example of an extension of a fragment of BPEL process with ontological semantic anotations. In this case, the semantization consists in choosing suitable domain ontologies and establishing links between ontology concepts represented by the means of the descriptive logic and metadata describing the functionality of the web service. The annotated system of transition states (ASTS) is the computer- understandable service presentation model. This is due to the fact that, in particular, most of the AI planning systems work with STS, the mechanisms of which can also be used to automatically solve web service tasks.
Key words: web-service, web-service problems, web-service semantization, semantics categories, process model, interface, pre-condition, post-condition, effect, semantic annotation, annotating process.

Вступ

Веб-севіси – це програмні компоненти, що доступні як веб-ресурси. Тому, розробка веб-сервісів з метою реалізації лише однією конкретної бізнес-задачі, не забезпечуючи їх подальшого повторного використання, є не ефективною та коштовною роботою. Для уможливлення повторного використання сервісу іншими сервісами або програмами необхідно забезпечити абстрактний опис сервісу, який міститиме базову технічну інформацію про нього, а саме про те: що сервіс робить, як його можна викликати, які параметри мають бути забезпечені для веб-сервіса. Щоб вбудувати сервіс у бізнес-процес організації, він також має містити інформацію про те, якому протоколу необхідно слідувати при використанні веб-сервісу. Користувач веб-сервісу може мати деякі очікування або вимоги щодо властивостей сервісу, які, наприклад, можуть стосуватися проблем безпеки.

Зрозуміло, що цей список можна продовжити, але він дає розуміння основних аспектів використання сервісу. Очевидно, що сторони, які планують кооперуватися за допомогою веб-сервісу, потребують його специфікацій, які дозволять розділяти та експлуатувати технічні та не технічні описи сервісу. Але, одразу слід зазначити, що, якщо ці описи не формалізовані, не стандартизовані та є суто текстовими документами, то їх використання є дуже складною задачею. Насьогодні розроблено ряд стандартів (SOAP, UDDI, WSDL-S [1], WS-Security [2], WS-Transaction [3], WS-BPEL [4], WS-Policy [5]), що спрощують використання веб-сервісів, але вони не охоплюють різноманітних аспектів семантики сервісу. Семантичні специфікації веб-сервісів є результатом процесу семантичного анотування та повинні визначати загальну структуру, яка інтегрує семантичні описи багатьох відповідних властивостей веб-сервісів.

Така семантична структура необхідна для вирішення різних задач веб-сервісів, наприклад, виявлення сервісу, та його композиції з іншими сервісами, а також надає можливість визначати в запиті складні вимоги до сервісу.

Мета даної роботи – це дослідження самого процесу створення семантичної специфікації сервісу, визначення його основних аспектів процесу: від джерел інформації до конкретних кроків процесу, а також визначення місця дескриптивних логік (ДЛ) в процесі семантизації як найвиразнішого формального засобу представлення семантик.

Цілі та основні аспекти анотування веб-сервісів

Щоб визначити основні аспекти анотування сервісів, необхідно дати відповіді на такі питання:

1) які існують джерела інформації, що містять інформацію релевантну для специфікації веб-сервісу?
2) які описи потрібні для керування веб-сервісом та його повторного використання?
3) яким чином повинні бути представлені метадані?
4) як може бути використаний семантичний опис сервісу?

Мета анотування веб-сервісу полягає у тому, щоб спростити керування сервісами в веб шляхом часткової автоматизації задач керування. Головною проблемою автоматизації задач веб-сервісів є те, що більшість інформації про веб-сервіси представлено у форматі, що є зрозумілим лише для людини, тобто у вигляді текстової документації. Таким чином, необхідне створення таких описів веб-сервісів, що можуть інтерпретуватися машиною та заповнять розрив між існуючою інформацією про сервіси та потрібною для вирішення задач. Семантичні описи надають семантичні метадані, які описують семантики елемента сервісу в форматі, що може інтерпретуватися машиною. Таким чином, за допомогою семантичних метаданих семантичний опис визначає властивості сервісу. Семантичні метадані базуються на онтологіях та зв'язують властивості веб-сервісу з відповідними концептами в онтології. Для представлення онтологій використовуються мови представлення онтологій, більшість яких базуються на ДЛ. Безперечна доцільність використання апарата ДЛ обумовлена також тим, що системи ДЛ забезпечують користувачів різними механізмами виводу, які виводять неявні знання з тих, що явно представлені, та більше того, насьогодні існує вже чимало реалізованих механізмів міркувань (резонерів) ДЛ, що готові до використання.

Вхідні джерела інформації

Процес анотування семантичного веб-сервісу у загальному випадку потребує вхідної інформації з різних джерел. Інформація про сервіс може бути, наприклад, зібрана з вихідного коду сервісу, з API документації та опису, з загальної текстової документації сервісу або з документації у WS-стандарті. Залежно від ступеня структурованості цих джерел, семантичні анотації можуть забезпечуватися в ручну (на вході – просто текстовий документ), напів-автоматично (WS-опис) або повністю автоматично (наприклад, вхідна інформація базується на Java-інтерфейсі). Але повністю автоматично можуть забезпечуватися лише входи-виходи веб-сервісу, семантика операцій веб-сервісу та його пост- і передумови можуть бути визначені лише в ручну.

Як основні джерела інформації можна виділити:

- документацію веб-сервісу. Містить обрану інформацію щодо веб-сервісу. В більшості випадків, інформація надається природньою мовою та не є формалізованою. Визначити актуальні властивості сервісу на основі таких текстів досить складно або, навіть, не можливо для комп'ютера;
- вихідний код сервісу. Містить всю інформацію про операції сервісу, включаючи всі входи та виходи. Інформація, що витягується з вихідного коду, різниться в різних мовах програмування. Це можуть бути типи вхідних та вихідних параметрів, інформація про функціональність веб-сервісу, передумови веб-сервісу та ефекти виконання операції сервісу;
- API описи. Складаються з оголошень інтерфейсів у вихідному коді та пояснень природньою мовою. На відміну від документації, API описи фокусуються на конкретних операціях веб-сервісу, а не на веб-сервісі в цілому. На відміну від вихідного коду, API опис може бути детальнішим та часто фокусується на релевантних методах;
- коментарі, що включені до вихідного коду, також можуть використовуватися як для визначення інформації про операції, так і про сервіс в цілому;
- специфікації на програмне забезпечення. Забезпечують ті самі типи інформації, що й API описи;
- різні правові документи. Наприклад, ліцензійна угода користувача (EULA). Укладаються природньою мовою, тому їх також складно інтерпретувати;
- фонові знання – це інформація, яка зазвичай не забезпечується разом з веб-сервісом але грає важливу роль в його семантизації. Наприклад, фонові знання – це знання з програмування, що необхідні для розуміння технічних аспектів, знання про предметну область та відповідні закони, що роз'яснюють специфічні властивості домена.

Семантики описів

Проаналізуємо, які типи інформації служать якій меті анотування у взаємозв'язку з якими типами семантик веб-сервісів. Визначимо 4 категорії семантик [7], а саме:

- *семантики даних* – включають семантики всіх входів та виходів веб-сервісу [8];
- *семантики протоколу* – включають семантики, які стосуються протоколу, що визначає залежності між веб-сервісами та їх операціями, наприклад, порядок виконання;
- *функціональні семантики* покривають всі семантики, що пов'язані з функціональністю веб-сервісу;
- *не функціональні семантики* – всі семантики не функціональних властивостей сервісу, тобто всіх властивостей веб-сервісу, що не пов'язані безпосередньо з його функціональністю (якість, безпека тощо).

Наведені категорії семантик використовуються для семантичної специфікації веб-сервісу. Таким чином, семантична специфікація містить всю інформацію, що потрібна для виявлення сервісу та його оголошення. Комбінація семантик даних з розміщенням сервісу та семантиками middleware протоколів є спеціальною підмножиною специфікації семантичного сервісу, що зветься семантичною специфікацією заземлення та визначає, як отримати доступ до сервісу.

Основні типи описів веб-сервісу та кроки процесу анотування

Розглянемо процес анотування [7] (рисунок), виходячи з того, що описи сервісів розробляються інженерами, які впроваджують сервіс. Вони обізнані у вихідному коді сервісу та мають достатні знання для його анотування.

Перший крок – *не семантичні анотації.* Входи для цього кроку отримаються з різних інформаційних джерел, таких як, документація веб-сервісів та API-описи, описи інтерфейсів операцій веб-сервісів тощо. Виходами є WSDL-файли та інші типи не семантично анотованих API-описів та документів.

Опис операції – транзакції, як правило, дуже стислий та не містить дуже значущої точної інформації. Для уможливлення повторного використання та виявлення веб-сервісів, потрібна інформація про інтерфейс веб-сервісу. Інтерфейс – це властивість веб-сервісу, що може бути дуже просто описана, так як містить всі методи, що забезпечує веб-сервіс, як кінцеву точку комунікації машина-з-машиною, та повідомлення, які приймають та повертають ці методи. Інтерфейс методу може бути описаний за допомогою WSDL. За допомогою WSDL розробник може надати детальний опис веб-сервісу щодо того, як викликаються операції у веб-сервісі та в якому форматі повертається відповідь. Ця інформація є достатньо деталізованою, щоб забезпечити підтримку доступу до веб-сервісів, але не охоплює інформацію про функціональність веб-сервісу, його виконання та інші дані, які необхідно отримати з інших джерел.

Не семантична анотація є напівавтоматичною та підтримується різними засобами. Наприклад, існує багато мов програмування, що підтримують генерацію WSDL-файлів.

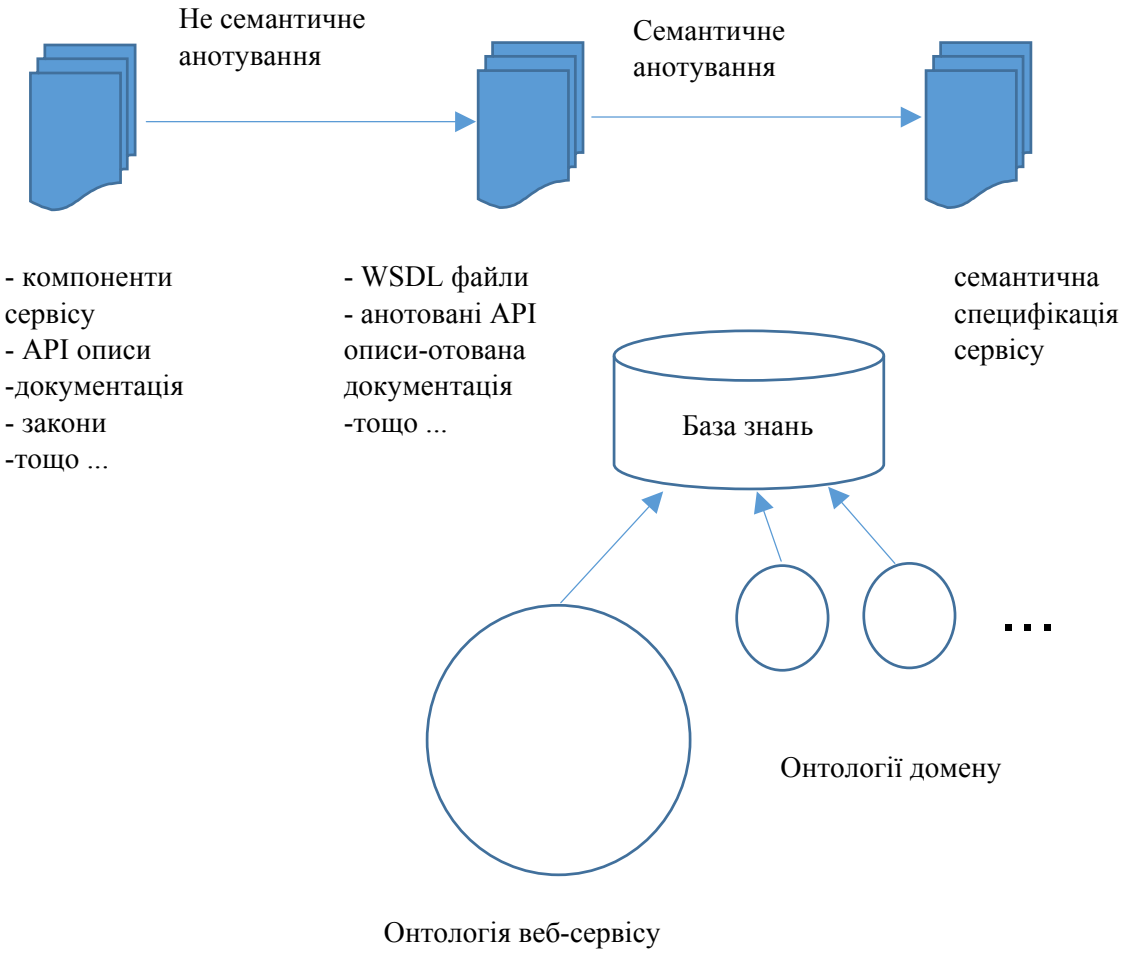

Рисунок. Процес анотування

Другий крок – *семантична анотація*. На цьому кроці приймаються такі самі входи, як і на першому, але в комбінації з релевантними онтологіями, такими як онтології веб-сервісів та різні онтології доменів. Виходом цього кроку є семантична специфікація сервісу.

Сам процес семантичної анотації також складається з декількох кроків.

Перший крок – це *класифікація* веб-сервісу. Результатом є призначення веб-сервісу конкретного прикладного домена. Розробник класифікує сервіс на основі власних знань або за допомогою засобів класифікації.

Після визначення класу прикладного домену веб-сервісу розробник має обрати відповідну онтологію домену, або, якщо такої не існує, визначити нову (можливо інтегровану на базі існуючих) за допомогою спеціальних засобів.

Щойно онтологію обрано, інженер повинен встановити зв'язки між метаданими, які описують функціональність веб-сервісу, та концептами онтології. Цей процес може бути поділений на різні підпроцеси відповідно до визначених вище типів семантик. Кожний підпроцес адресований одному типу семантик. Навіть, якщо не існує ніякого спеціального порядку виконання цих підпроцесів, є сенс розпочати з анотування метаданих семантиками даних. Щоб визначити семантики даних операції веб-сервісу, необхідно побудувати зв'язки між концептами онтології та відповідними параметрами входу й виходу, а також типами параметрів. Ці зв'язки визначаються шляхом визначення відповідних концептів онтології для кожного параметра та кожної операції веб-сервісу.

Для специфікації кінцевих точок веб-сервісів було розроблено WSDL [6], але йому не вистачає можливостей для вираження різних типів семантик. Надати описам веб-сервісів формальну семантику дозволяють онтологічні мови представлення знань, так як вони уможливлюють визначення концептів, клаузів, типів та зв'язків для специфікації, відповідно, концептуалізацій або онтологій. Якщо існують й не семантичні описи інтерфейсів, вони також можуть бути трансльовані в мову опису знань, що використовується для визначення семантичної специфікації. Залежно від мови, що використовується для не семантичного опису, ця трансляція може бути певною мірою автоматизованою. Наприклад, WSDL-файли

транслюються напівавтоматично внаслідок їх машинно-інтерпретаційній формі. Для такої трансляції типи, що використовуються, та елементи інтерфейсів операцій повинні бути відображені у концепти онтології. Результатом є семантична специфікація заземлення та частини передумов і ефектів, що специфікують вхідні та вихідні параметри.

Наступним кроком є *специфікація протоколу та функціональних семантик*. Семантики протоколу описують протокол, якому необхідно слідувати для вирішення бізнес задачі за допомогою сервісу. Протокол визначає порядок операцій, включаючи послідовність, петлі, розгалуження, з'єднання тощо. У загальному випадку семантики протоколу описують робочі потоки та потоки даних релевантні виконанню веб-сервісу. Протокол може визначатися в різних нотаціях, що більш-менш інтуїтивно зрозумілі для розробника. У подальшому, семантики протоколу можуть специфікуватися за допомогою передумов та постумов. Функціональні семантики описують функціональність операцій веб-сервісу. Вони можуть також виражатися за допомогою передумов, постумов та ефектів.

Розрізняють внутрішні передумови, які перевіряються (внутрішньо) прикладною логікою або забезпечуються умовами, що накладаються на типи мовою програмування, та зовнішні – більш загальні передумови, що повинні задовольнятися для коректного та успішного виконання веб-сервісу та не можуть бути перевірені в межах реалізації сервісу. Далі, у внутрішніх умовах розрізняють явні та неявні умови. Неявні передумови – це умови, що випливають зі стандартів, типів та визначень. Наприклад, «дата повинна бути представлена у конкретному форматі». Явні передумови – це умови, що визначаються для конкретної операції. Наприклад, термін дії кредитної карти.

Ефекти описують вихід та вплив виконання веб-сервісу. Якщо термін постумова часто фокусується лише на описі умов, що стосуються значень, які повертають операції веб-сервісу, ефект розглядає також вплив, який має виконання веб-сервісу. Наприклад, ефектом переказу грошей є зміна стану рахунку.

Якщо визначені семантики даних, то частини протоколу та функціональність вже неявно специфіковані. Неявна специфікація протоколу випливає зі специфікації вхідних параметрів. З цієї специфікації можна зробити висновок, яку інформацію необхідно зібрати до виклику операції. Неявна специфікація функціональності є частиною специфікацій параметрів. Іншими словами, всі частини протоколу та функціональності, що можуть бути виведені з потоків даних, можуть бути неявно специфіковані. Інші частини семантик потребують явної специфікації за допомогою встановлення зв'язків. Входом для цього кроку процесу є інформація з усіх джерел, що описувались раніше. Явні специфікації передумов та ефектів є складними та повинні визначатися розробником вручну.

Останній крок семантичної анотації – *специфікація не функціональних семантик* веб-сервісу. Існує багато різних не функціональних властивостей: наприклад, вартість веб-сервісу, безпека (комунікація, шифрування даних), якість сервісу (показник якості, у свою чергу, може охоплювати багато факторів: час виклику операції, можливості сервісу, інформація про використання ресурсів веб-сервісу). Щоб досягнути коректної інтерпретації не функціональних властивостей, необхідно знайти відповідні описи. Деякі з простих не функціональних властивостей таких, як назва сервісу, можуть бути автоматично витягнуті з WSDL файлу. Більш складні не функціональні властивості описуються інженером. Для цього можуть бути використані різні джерела інформації, наприклад, стандарти або закони. Так, законом може регламентуватися мінімальний рівень шифрування для передачі специфічних даних.

Загальними аспектами процесів не семантичної та семантичної анотації є зусилля анотування. Опис сервісу, що містить всі релевантні властивості, є суттєвим важелем впливу для його повторного використання та уможливлює кращу підтримку засобу, але, в свою чергу, вимагає від інженера більше зусиль. При вирішенні даної проблеми повинен досягатися компроміс між зусиллями моделювання та менеджменту [9], де зусилля моделювання охоплюють процеси анотації та документації, в той час як зусилля менеджменту означають зусилля, які необхідно прикласти менеджеру для забезпечення повторного використання сервісу. Зменшення зусиль на анотування призводить до збільшення зусиль менеджменту та навпаки. Розумний компроміс, зазвичай, досягається при мінімальній комбінації, але для різних сценаріїв є інші фактори, що впливають на оптимальну комбінацію. Наприклад, провайдер сервісу може збільшити свої зусилля моделювання, щоб зменшити зусилля менеджменту, та отримати перевагу серед конкурентів. Таким чином, особа, яка відповідає за анотування сервісу повинна знайти власний компроміс залежно від індивідуального контексту.

Загалом, семантизація сервісу окрім анотування передбачає подальше *конвертування* моделі. Річ у тім, що навіть добре анотована семантична специфікація сервісу не забезпечує автоматизованого вирішення задач веб-сервісів, яка потребують урахування їх поведінки. Тому, потрібна така модель представлення, яка може бути перевірена легко і автоматично. Вираження поведінкової моделі сервісу в такій формі і є суттю етапу **конвертування моделі**. Зокрема, оскільки контролери моделі зазвичай мають справу з деяким видом систем переходів станів, є доцільним переведення анотованого опису процесу в анотовану систему перехідних станів (ASTS). Цей крок може виконуватися автоматично.

Семантичне анотування на прикладі розширення процедурного опису сервісу

Поведінкові аспекти сервісу можуть бути представлені різними мовами. Для демонстрації прикладу представлення процедурного опису веб-сервісу будемо використовувати BPEL. Композиція WSDL+BPEL є «класичним», чисто синтаксичним поданням процесу та забезпечує потужний інструментальний засіб вирішення проблем.

BPEL [9] 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 та 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. У загальному випадку BPEL специфікація дуже детально визначає шлях взаємодії, який необхідно проводити з веб-сервісом для того, щоб його використовувати, але, цього ще не достатньо, щоб дозволити автоматичне вирішення задач веб-сервісів, наприклад, просто виявлення сервісів. Для цього BPEL специфікації повинні бути розширені «семантичними анотаціями». Анотування процесної моделі сервісу здійснюється введенням в BPEL специфікацію спеціальних атрибутів semann, а їх значеннями є семантичні анотації.

Як приклад, розглянемо сервіс бронювання квитків на літак. Його BPEL специфікація може мати вигляд:

```
<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 : POOntology#Location, /req/dest : POOntology# Location, /req/date :
    POOntology#Date"/>
        <switch name="checkAvailability">
          <case name="isNotAvailable">
            <invoke operation="not_avail" partnerLink="client" semann="/offer/fl : POOntology#Trip,
        /offer/fl.status = notAvailable"/>
          </case>
          <otherwise name="isAvailable">
            <assign name="prepareOffer">
            <copy><from opaque="yes" semann="/offer/fl : POOntology#Trip, /offer/fl.start = /req/start,
                /offer/fl.destination = /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/client : POOntology#Client, /offer/fl.pax = /client/pax,
            /offer/flight.status = booked"/>
```

```
                    <onMessage operation="nack" partnerLink="client" semann="/offer/flight.status =
                cancelled"/>
                    </pick>
                </otherwise>
            </switch>
            </sequence>
</process>
```

Представлений фрагмент визначає приклад семантичної специфікації можливого інтерфейсу для сервісу резервації квитків на літак – сервіс *FlightReservation*. Після отримання запиту на квиток (операція *receive*), сервіс процесу приймає рішення про наявність квитків (операція switch названа *checkAvailability*). Зауважимо, що реалізація цієї перевірки, яка залежить від інформаційної системи авіакомпаній, не має відношення до опису протоколу взаємодії і, отже, не зазначена в абстрактному BPEL. Якщо квиток не доступний (case *isNotAvailable*), це доводиться до клієнта і сервіс закінчує виконання. Якщо квиток доступний (case *isAvailableTickets*), сервіс відправляє пропозицію (*invoke Offer*) і, в разі, якщо клієнт її приймає (*OnMessage* операція *Ack*), квиток бронюється (*flight.status = booked*), в іншому випадку (*OnMessage* операція *NAck*), операція скасовується (*flight.status = cancelled*).

Наведений приклад демонструє встановлення зв'язків між концептами онтології домену *POOntology* та вхідними і вихідними повідомленнями, якими обмінюється процес. Це досягається через використання семантичних анотацій *"/req/start : POOntology#Location, /req/dest : POOntology# Location, /req/date : POOntology#Date"*. Окрім цього, виражаються "семантичні" відношення між значеннями вхідних та вихідних даних, переданих під час взаємодії з веб-сервісом, визначаються результати взаємодії з веб-сервісом

(*"/offer/fl : POOntology#Trip, /offer/fl.start = /req/start, /offer/fl.destination = /req/dest, /offer/fl.date = /req/date"*).

У наведеному прикладі ***POOntology –*** онтологія прикладного домену, що представлена за допомогою ДЛ. Її узагальнений ТВох може містити наступні твердження:

Year, Month, Day, Date, Client, Status, Location, Country, FlightID.

Date □ has.Year; Date □ has.Month; Date □ has.Day; Location□ isLocatedIn. Country; Trip □ hasStartPoint. Location;Trip □ hasEndPoint. Location; Trip □ hasDeparture.Date; Trip □ hasArrival.Date; Flight□ belonguesTo.AirLineRoute; Flight □ hasDeparturDate.Date; Flight □ hasDeparturTime.Time; Flight □ from.Location; Flight □ to.Location; Flight□ hasSeats.NUMBER; Flight□ hasStatus.Status; Status = {Available, NotAvailable, booked}.

Наведена онтологія містить як загальні, незалежні від домену концепти (Day, Month, Year, …), так і концепти, що є специфічними для даного домену (Location, Trip, …). BPEL специфікацію можна умовно розділити на дві частини: декларація змінних, які використовуються у вхідних/вихідних повідомленнях, та решта, що описує потік взаємодії.

Визначимо *основні аспекти використання семантичних анотацій* в описі сервісу:

- забезпечення зв'язків між поняттями онтології та вхідними і вихідними повідомленнями, якими обмінюється процес. Цю роль відіграють анотації *«/req/start : POOntology#Location, /req/dest : POOntology# Location, /req/date : POOntology#Date»* діяльності receive для операції request на початку опису;
- вираження «семантичних» відношень між значеннями вхідних і вихідних параметрів, якими обмінюються веб-сервіси при взаємодії. Це визначається в анотації *«/offer/fl.start = /req/start, /offer/fl.destination = /req/dest, /offer/fl.date = /req/date»*;
- визначення результату взаємодії з веб-сервісом. В даному прикладі, рейс бронюється лише за умови його доступності. Сервіс відсилає пропозицію і користувач визначає згоду з пропозицією. Для цього в діяльність, що відповідає за отримання підтвердження, додається анотація *«/offer/status = booked»*.

Таким чином, семантичні анотації потрібні, щоб корегувати особливості інтерфейсу, і визначати це у відношенні із загальною онтологією.

Семантизація STS. Місце ДЛ у визначенні ASTS -моделі

Метою семантичного анотування опису веб-сервісів є уможливлення їх повторного використання та подальшого автоматизованого та ефективного вирішення їх задач. Тому, наступним етапом семантизації, за наявності анотованого опису моделі сервісу, є її конвертування, а саме перетворення її у форму, що може оброблятися автоматично. Такою формою є системи перехідних станів (STS). З STS, зокрема, працюють більшість систем AI планування, механізми яких можуть також використовуватися для автоматичного вирішення задач веб-сервісів.

BPEL процеси, розширені семантичними анотаціями, кодуються як ASTS. STS описують динамічні системи, які можуть бути в одному з їх можливих станів (деякі з них позначені як вихідні стани), і можуть розвиватися до нових станів в результаті виконання деяких дій. Всі дії поділяються на вхідні, вихідні і τ. Вхідні дії – це прийом повідомлень, вихідні – представляють повідомлення, передані до зовнішніх сервісів, а τ є спеціальною дією, що називається внутрішня дія та представляє внутрішні зміни, які не видно зовнішнім сервісам. Іншими словами, τ представляє той факт, що стан системи може розвиватися, не виробляючи будь-якого виходу та не споживаючи будь-якого входу. Це є наслідком того, що в абстрактному BPEL внутрішні дії «непрозорі». Відношення переходу описує, як стан може розвиватися на основі входів, виходів або внутрішньої дії τ.

В ASTS з кожним станом можна асоціювати набір тверджень концептів та тверджень ролі. Це конфігурує стан як стверджувальний компонент (ABox) системи подання знань на основі ДЛ, де сама онтологія грає роль термінологічного компонента (TBox). У такому разі, кожний ABox характеризує окремий стан системи. Як наслідок, кожну дію можна розглядати як перехід зі стану, що знаходиться в ABox, в інший стан, що міститься в іншому ABox. Слід зазначити, що для анотованих BPEL процесів така конвертація у ASTS може виконуватися автоматично.

Визначимо роль ДЛ у ASTS –представленні сервісів. Перш за все, розглянемо загальне визначення ASTS.

Визначення [10]. ASTS, яка визначена на системі перехідних станів Σ, – є кортежем (Σ, T, Λ), де: $\Sigma = \langle S, S^0, I, O, R, P, X \rangle$ – це система перехідних станів; S – кінцевий набір станів; $S^0 \subseteq S$ – набір початкових станів; I – кінцевий набір вхідних дій; O – кінцевий набір вихідних дій; $R \subseteq S \times (I \cup O \cup \{\tau\}) \times S$ – відношення переходу; P – набір висловлювань, який в ASTS є порожнім (P = 0); $X : S \rightarrow 2^P$ – функція спостереження, яка в анотованій STS є невизначеною; T – термінологія анотації (TBox); $\Lambda : S \rightarrow 2^{A_T}$ – функція анотування, де A_T – набір всіх тверджень концептів і тверджень ролей, визначених на T.

Розглянемо приклад фрагменту ASTS для процесу *FlightReservation,* BPEL – опис якого наведений вище. Перш за все, слід зазначити, що в якості термінології анотацій використовується TBox онтології *POOntology*.

PROCESS *FlightReservation*;

STATE *pc : { START, ReceiveRequest, CheckAvailability, isNotAvailable, invokeNot_Avail, isAvailable, PrepareOffer, invokeOffer, pickWaitAcknowledge, requestAck, Ack, END_PICK, END_ACK, END_NACK };*

INIT *pc = {START};*

CONCEPT *String; Client; Number; Date; Location; Flight; Trip, Country, FlightID; AirLineRoute; Status;*

INPUT *request(flightRequestMsg); NAckMsg(); AckMsg();*

OUTPUT *isNotAvailable(); waitAcknowledge(Status)*;

TRANS

pc = START -[TAU]-> *pc = ReceiveRequest*;

pc = ReceiveRequest -[INPUT request(flightRequestMsg)]-> pc = *CheckAvailablility*

pc = CheckAvailability -[TAU]-> *pc = isNotAvailable*;

pc = CheckAvailability -[TAU]-> *pc = isAvailable*;

...

ANNOTATION FUNCTION

...

LAMBDA(*ReceiveRequest*) = LAMBDA(*START*);

LAMBDA(*CheckAvailability*) = { */flightRequestMsg/client:Client,*
/flightRequestMsg/start:Location,
/flightRequestMsg/des:Location,
/flightRequestMsg/date:Date}

U LAMBDA(ReceiveRequest);

LAMBDA(*PrepareOffer*) = { */flightOfferMsg/client:Client,*

/flightOfferMsg/start:Location,

/flightOfferMsg/des:Location,

/flightOfferMsg/date:Date};

…

Запропоноване представлення є фрагментом текстового опису ASTS, що відповідає BPEL процесу прикладу. Відповідно до формального визначення ASTS, набір станів *S* (розділ *STATE*) моделює етапи процесу та еволюцію тверджень поняття і ролі. *pc* є змінною, яка пробігає набір станів *S* і, отже, фіксує поточний крок виконання сервісу. Набір початкових станів S_0 представлено розділом *INIT*. Концепти, що використовуються в анотованій STS, перераховані в розділі *CONCEPT*. Вони повинні бути визначені в термінах TBox-а □□онтології *POOntology*. Відповідно до формальної моделі, розрізняють три різні види дій. Вхідні дії *I* моделюють всі вхідні запити до процесу та інформацію, яку вони приносять. Вихідні дії *O* моделюють всі дії, які відправляють вихідні повідомлення. Дія τ використовується для моделювання внутрішніх еволюцій процесу, таких як призначення й прийняття рішень. Набір вхідних дій *I* (вихідних дій *O*) представлено розділом *INPUT* (*OUTPUT*). Еволюція процесу моделюється у розділі *TRANS*. Функція анотації *Λ* (розділ *ANNOTATION FUNCTION*) моделює, як твердження варіюються залежно від станів. Наприклад, *LAMBDA(PrepareOffer) = { /flightOfferMsg/Client:Client, /flightOfferMsg/start:Location, /flightOfferMsg/des:Location, /flightOfferMsg/date:Date};* представляє той факт, що стан *PrepareOffer* містить, наприклад, твердження концепту *flightOfferMsg/Client:pax, /flightOfferMsg/start:Location, /flightOfferMsg/des:Location, /flightOfferMsg/date:Date* є індивідами, які належать до концептів *Client*, *Location* та *Date*, відповідно. Кожна клауза *TRANS* і кожна клауза *LAMBDA* відповідає різним елементам у відношенні переходу □□ і в функції анотації *Λ*, відповідно.

Для того, щоб отримати конкретну систему перехідного стану (набір конкретних ABoxes), концептам TBox має бути призначений кінцевий набір індивідів цих концептів. Можливий підхід – визначити відповідні твердження концептів у загальній частині ABoxes. Інший, кращий метод полягає у використанні засобів рівня знань, таких, як в [11], щоб уникнути явного переліку індивідів.

Щоб перевірити, чи є істинним твердження p в даному стані, застосовується перевірка екземпляра, що позначається як $\langle \square\square, \Lambda(s)\rangle \models p$ [10]. У конкретному випадку перевірки категоризації, ABoxes не грають активної ролі [12], тому категоризація може бути перевірена не враховуючи, яким є поточний стан (тобто поточний ABox). Наприклад, коли ми повинні перевірити $\langle \square\square, \Lambda(s)\rangle \models C \sqsubseteq D$, нам потрібно тільки перевірити $\langle \square\square, \emptyset\rangle \models C \sqsubseteq D$.

Так як, ми акцентували увагу на використанні апарату ДЛ в процесі семантичного анотування сервісу, то слід зазначити, що існує окрема досить складна проблема, яка потребує вирішення. Це вибір мови з сімейства ДЛ, досить виразної для опису нетривіальних прикладів, яка б дозволила формалізувати складний процес веб-сервісу, але з допусти мою обчислювальною складністю вирішення таких задач, як категоризація та перевірка екземпляра.

Висновки

Для забезпечення можливості повторного використання сервісів, вони повинні бути семантизовані, тобто збагачені семантичними анотаціями, які охоплюють різні аспекти та властивості сервісу. Складні бізнес-задачі вимагають використання веб-сервісів не лише як атомних одиниць, що мають тільки входи, виходи, передумови та ефекти, а з урахуванням їх поведінкових особливостей. Але звичайна процесна модель, що представлена за допомогою однієї з процедурних мов, досить детально описує сам процес як послідовність дій, але не дає можливості автоматизованого використання такого сервісу для вирішення інших задач в подальшому. Для цього формальний опис сервісу повинен бути збагачений семантичними описами, зрозумілими комп'ютеру. Насьогодні більшість існуючих сервісів не семантизовані, а джерела інформації, що містить необхідну інформацію, не формалізовані достатньою мірою. У зв'язку з цим потребує детального аналізу та специфікації сам процес анотування опису веб-сервісу та подальшого його приведення до форми, що є зрозумілою для комп'ютера. Запропонований покроковий алгоритм семантизації сервісу дозволяє більш формально визначити сам процес анотування, описує його основні аспекти (зміст, входи, виходи кожного кроку, категорії необхідних семантик) та виявляє головні проблеми.

Анотування певних категорій семантик, в першу чергу, семантик даних, виконується шляхом встановлення зв'язків з концептами відповідних онтологій. ДЛ є формальною мовою, що підтримує концепцію відкритого світу та власні механізми міркувань. Більшість задач, для вирішення яких можуть застосовуватись

резонери ДЛ, зводяться до стандартної задачі виконуваності. Все це робить її ефективною для формального визначення онтологій, що використовуються для семантичної специфікації сервісу, та подальшого визначення функціональної частини опису веб-сервісу та семантичних елементів у процедурному описі веб-сервісу.

Література

1. Akkiraju R., et al. (2005, December 6). Web Service Semantics - WSDL-S. Available: http://www.w3.org/Submission/WSDL-S/
2. https://msdn.microsoft.com/en-us/library/ms977327.aspx
3. http://searchmicroservices.techtarget.com/definition/WS-Transaction
4. https://www.ibm.com/developerworks/webservices/tutorials/ws-understand-web-services7/index.html
5. Http://www.w3.org/Submission/WS-Policy/
6. WSDL - http://www.w3.org/TR/wsdl
7. Christoph Ringelstein, Thomas Franz, Steffen Staab. The Process of Semantic Annotation of Web Services./ ISWeb, University of Koblenz-Landau, Germany – http://isweb.uni-koblenz.de.
8. Patil A.A., Oundhakar S.A., Sheth A.P., and Verma Kunal. METEOR-S Web Service Annotation Framework, WWW 2004, ACM Press. 2004. P. 553–562.
9. Oberle D. Semantic Management of Middleware, New York, USA: Springer, 2006.
10. 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.
11. Pistore M., Marconi A., Bertoli P., and Traverso P. Automated Composition ofWeb Services by Planning at the Knowledge Level. In Proc. IJCAI'05, 2005.
12. Schaerf A. Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues. Dottorato di Ricerca in Informatica, Universit`a degli Studi di Roma "La Sapienza", Italia. 1994.

References

1. Akkiraju R., et al. (2005, December 6). Web Service Semantics - WSDL-S. Available: http://www.w3.org/Submission/WSDL-S/
2. https://msdn.microsoft.com/en-us/library/ms977327.aspx
3. http://searchmicroservices.techtarget.com/definition/WS-Transaction
4. https://www.ibm.com/developerworks/webservices/tutorials/ws-understand-web-services7/index.html
5. Http://www.w3.org/Submission/WS-Policy/
6. WSDL - http://www.w3.org/TR/wsdl
7. Christoph Ringelstein, Thomas Franz, Steffen Staab. The Process of Semantic Annotation of Web Services./ ISWeb, University of Koblenz-Landau, Germany – http://isweb.uni-koblenz.de.
8. Patil A.A., Oundhakar S.A., Sheth A.P., and Verma Kunal. METEOR-S Web Service Annotation Framework, WWW 2004, ACM Press. 2004. P. 553–562.
9. Oberle D. Semantic Management of Middleware, New York, USA: Springer, 2006.
10. 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.
11. Pistore M., Marconi A., Bertoli P., and Traverso P. Automated Composition ofWeb Services by Planning at the Knowledge Level. In Proc. IJCAI'05. 2005.
12. Schaerf A. Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues. Dottorato di Ricerca in Informatica, Universit`a degli Studi di Roma "La Sapienza", Italia, 1994.

Про автора:

Захарова Ольга Вікторівна,
кандидат технічних наук,
старший науковий співробітник.
Кількість наукових публікацій в українських виданнях – 27.
http://orcid.org/0000-0002-9579-2973.

Місце роботи автора:

Інститут програмних систем НАН України,
проспект Академіка Глушкова, 40.
Тел.: 526 5139.
Моб. тел.: +38 (068) 594 7560.
E-mail: ozakharova68@gmail.com.