

ТЕЛЕНИК С.Ф.,
АМОНС О.А.,
ТРОЦЬКИЙ С.О.,
БІЛОРУС Д.С.

АВТОМАТИЗАЦІЯ ПІДТРИМКИ БІЗНЕС-ПРОЦЕСІВ У WEB-СЕРЕДОВИЩІ

У статті розглянуто основи функціонування і реалізації системи автоматизації підтримки бізнес-процесів, яка дозволяє значно спростити розробку та підтримку бізнес-процесів підприємства за рахунок впровадження сервіс-орієнтованої архітектури та виконання системних завдань за допомогою наявних у Web-середовищі сервісів. Система підбирає комплекс необхідних для підтримки бізнес-процесу Web-сервісів та надає єдину точку входу для взаємодії з ними системним завданням.

The subject of the article is basic activities of functioning and implementation of business process automation support system that significantly streamline the development and support of business processes by implementing a service-oriented architecture and perform system's tasks with using services which is currently exist in web environment. The system searches and selects complex of Web-services to support business process and provides a single entry point to interact with them by system's tasks.

Вступ

В основі організації діяльності підприємств лежать бізнес-процеси. Інтегровані у єдину систему вони визначають поведінку підприємства за рахунок координованої взаємодії співробітників, спрямованої на визначення і ефективну реалізацію стратегію в умовах мінливого бізнес-середовища. Для автоматизації бізнес-процесів використовують спеціальні інформаційні системи виконання бізнес процесів. Використання сервіс-орієнтованої архітектури в таких системах дозволяє швидко адаптуватися до мінливих умов бізнес-середовища. В свою чергу, автоматичний пошук та композиція існуючих Web-сервісів може значно полегшити впровадження та підтримку цих систем. Розробленню моделей і методів співставлення операцій бізнес-процесу та існуючих Web-сервісів, їх вибору і виконанню і присвячена стаття.

Огляд проблеми

У загальному випадку бізнес-процес – це сукупність взаємопов'язаних заходів і завдань, спрямованих на створення певного продукту або послуги для споживачів. Бізнес-процеси виконуються всередині кожної організації, незалежно від того, формалізовані вони чи ні. Популярна сьогодні система умовних позначень (нотація) BPMN визначає скінченний набір графічних елементів для описання бізнес-процесу [1]:

- Об'єкти потоку управління: події, дії та логічні оператори;
- З'єднувальні об'єкти: потік управління, потік повідомлень та асоціації;
- Ролі: пули і доріжки;
- Артефакти: дані, групи і текстові анотації.

Найменшою елементарною дією в бізнес-процесі є завдання (task). З точки зору автоматизації бізнес-процесів, саме завдання є головним цільовим об'єктом. Інші елементи пов'язані з розв'язанням задач оркестрування, хореографії, координування, та ін. Для цих задач успішно застосовують мови виконання бізнес-процес, наприклад, BPEL.

На Рис. 1 у нотації BPMN 2.0 зображений спрощений узагальнений бізнес-процес. Він складеться з двох учасників: деякого програмного забезпечення (Customer system) та оператора (Operator), що взаємодіє з ним.

У наведеному процесі дії, що автоматично виконуються системою (Auto-task) межують з діями, що потребують втручання людини (User task). При цьому, логіка виконання системних дій є невід'ємною частиною програмного забезпечення, що відповідає за виконання бізнес-процесу. Процес, побудований таким чином, може добре виконувати покладені на нього функції та має як переваги, насамперед, простоту та повний контроль над системою, так і недоліки, серед яких найбільш суттєвими є статичність процесу, висока складність розробки, складність підтримки та модифікації, склад-

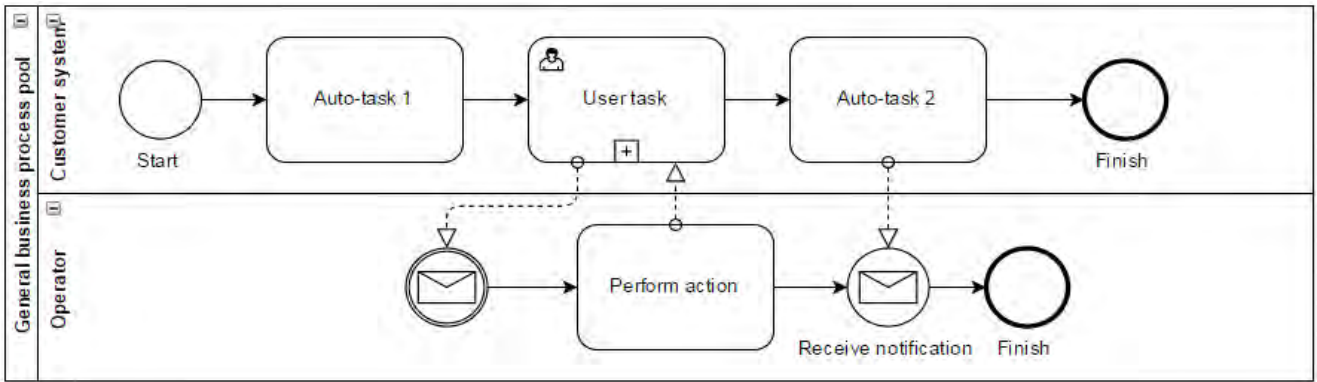


Рис. 1. Приклад бізнес-процесу

ність повторного використання логіки виконання системних дій.

Частково усунути недоліки описані вище можна за рахунок впровадження сервіс-орієнтованої архітектури. Сервіс-орієнтована архітектура (Service-Oriented Architecture, SOA) становить собою стиль створення архітектури ІТ, спрямований на перетворення бізнесу в ряд пов'язаних сервісів – стандартних бізнес-задач, які можна при необхідності викликати через мережу [2]. Це може бути Інтернет, локальна мережа або географічно розподілена мережа, яка об'єднує різні технології і поєднує сервіси так чином, ніби вони встановлені на одній локальній машині. Для виконання певного бізнес-завдання можна об'єднувати множину таких сервісів. Це дозволяє компаніям швидко адаптуватися до змін умов і вимог ринку.

Визначимо Web-сервіси як автономні, модульні програми, призначені для реалізації бізнес-процесів. Побудова і надання Web-сервісів спираються на ряд галузевих стандартів: WSDL (для описання сервісів), UDDI (для інформування та публікації); SOAP (для обміну повідомленнями) [3]. Ці специфікації не залежить від

платформи і мови, завдяки чому користувачі можуть пов'язувати різні компоненти з різних організаційних структур. Розглянемо сучасний підхід, при якому існуючий бізнес процес буде виконуватися за допомогою інформаційної системи управління бізнес-процесами. А сам бізнес процес, за виключенням логічних операцій та ручних дій, має підтримуватися існуючими Web-сервісами, задовольняючи основоположні принципи SOA. На Рис. 2 зображено модифікований бізнес-процес, що задовольняє умови описаного підходу та характеризується такими перевагами:

- розділення логіки управління бізнес-процесу та бізнес-логіки, що додає гнучкості системи та зменшує вартість підтримки;
- можливість розробки функціоналу бізнес-логіки зручними для кожного окремого випадку способами;
- можливість використання Web-сервісів третіх сторін, що зменшує час на розробку системи;
- можливість повторного використання Web-сервісів у різних частинах та процесах системи.

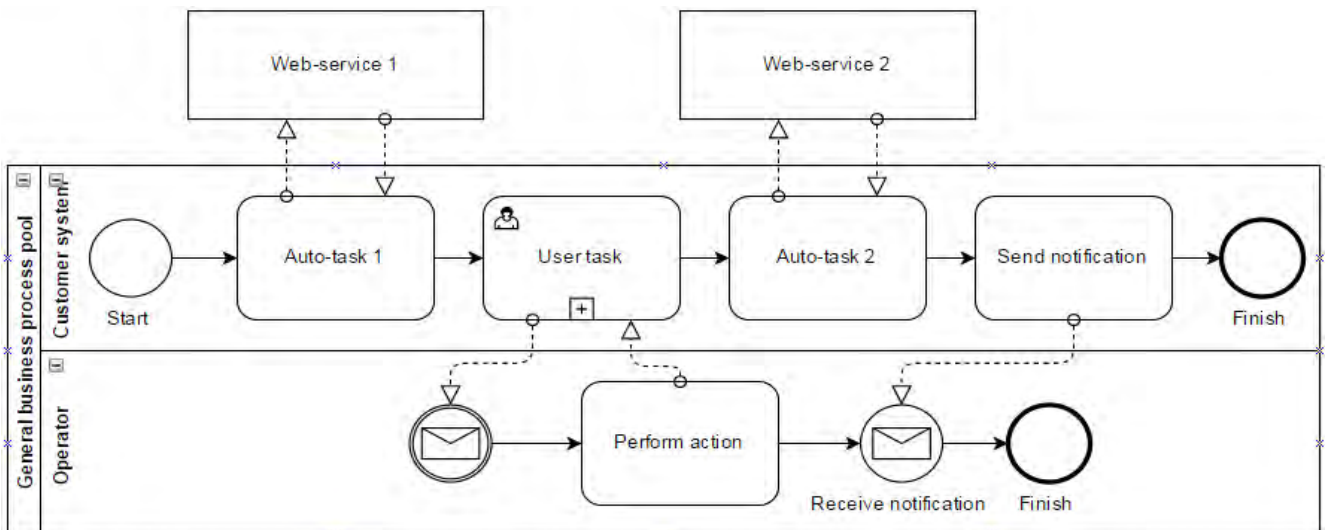


Рис. 2. Бізнес процес з використанням SOA

В даному процесі логіка виконання системних дій винесена у Web-сервіси, що можуть бути реалізовані будь-якою мовою програмування та за допомогою будь-яких технологій. В основному потоці при цьому лишаються користувацькі дії та дії виклику Web-сервісів. Таким чином, ми можемо повністю перекласти виконання основного потоку на один з серверів виконання бізнес-процесів. Промисловим стандартом у даній галузі є Oracle BPEL Process Manager Server [4], що є сервером виконання бізнес-процесів, описаних мовою виконання бізнес-процесів BPEL (Business Process Execution Language).

Головним недоліком, описаного вище сучасного і досить розповсюдженого підходу є необхідність ручного пошуку Web-сервісів та їх жорстка прив'язка до кожної дії бізнес процесу. При цьому, за будь-яких змін у структурі Web-сервісу або виходу його з ладу – вся система буда непрацездатною. Для уникнення такої ситуації і надання більшої незалежності системам управління бізнес-процесів підприємства автори пропонують побудувати окрему систему (далі за текстом – система автоматизації підтримки бізнес-процесу), яка візьме на себе задачу пошуку та композиції Web-сервісів для підтримки будь-якого бізнес-процесу та зведе до мінімуму необхідність ручного втручання в бізнес-логіку процесу.

запропонований підхід, який дозволяє вирішувати проблему підтримки бізнес-процесів за рахунок існуючих технологій і легко адаптуватися до нових, насамперед, семантичних технологій організації Web-сервісів.

Беручи до уваги переваги та недоліки існуючих підходів, автори розробили систему підтримки бізнес-процесів, що буде єдиною точкою входу для взаємодії з зовнішніми Web-сервісами. Тобто, замість звернення безпосередньо до зовнішніх Web-сервісів, сервер виконання взаємодіє лише з системою автоматизації бізнес-процесу, що значно спрощує опис бізнес-процесу. При цьому, відповідальність за підбір та забезпечення працездатності Web-сервісів повністю лягає на систему автоматизації. Для збереження усіх переваг COA, система автоматизації має надавати інтерфейс взаємодії у вигляді Web-сервісу. Приклад бізнес-процесу, що використовує дану систему зображений на Рис. 3.

Система, побудована таким чином, успішно вирішує такі завдання:

- Аналіз бізнес-процесу, виявлення системних дій;
- Динамічна генерація методів Web-сервісу у відповідності до описаних системних дій;
- Пошук реалізацій завдань бізнес-процесу;
- Вибір оптимальної реалізації серед множини можливий реалізацій;
- Забезпечення зв'язку між згенерованими методами та їх реалізаціями.

Відповідно до підходу, що пропонується, си

Побудова системи автоматизації підтримки бізнес процесів

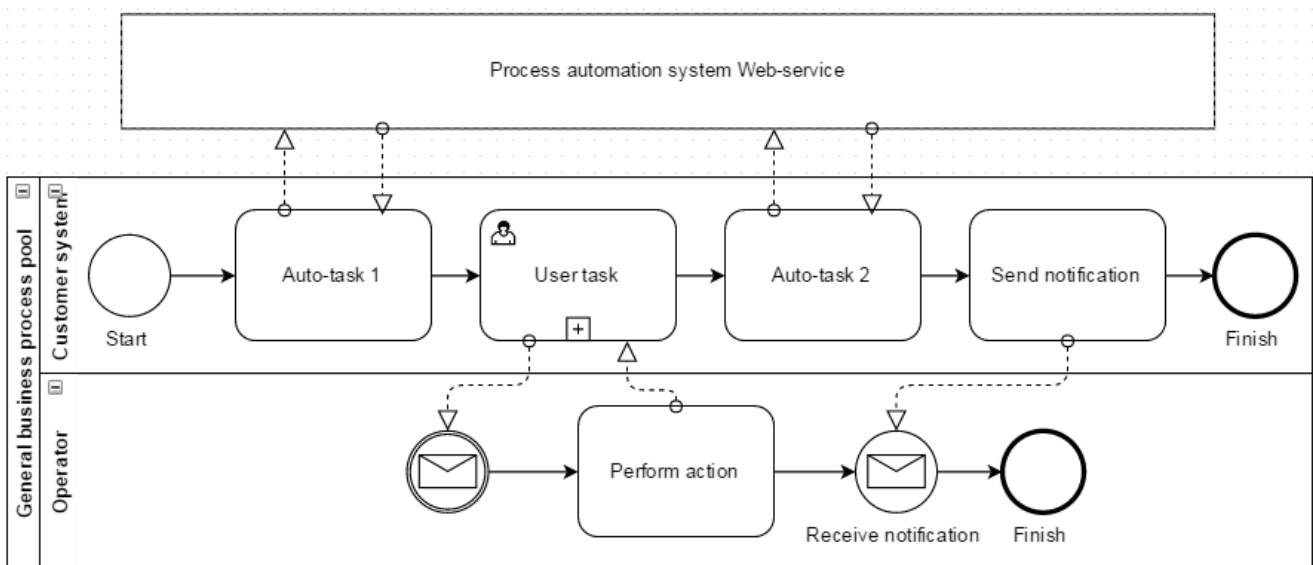


Рис. 3. Бізнес-процес з використанням системи автоматизації

Виходячи із реалій сьогодення, але й враховуючи перспективи розвитку ІТ-галузі, в роботі

система автоматизованої підтримки буде працювати у таких режимах:

1. Базовий – використовуючи засоби автоматизації роботи з нотацією для моделювання бізнес-процесів BPMN, оператор вибудовує схеми бізнес-процесів, визначаючи елементи, які система буде автоматично підтримувати за рахунок явно визначених Web-сервісів;

2. Автоматизований (реальний) – система, на основі попередньо визначеного оператором опису бізнес-процесів, здійснює пошук Web-сервісів під кожний елемент, що може бути виконаний автоматично. Серед знайдених Web-сервісів оператор відбирає ті, що дійсно задовольняють необхідні потреби, після чого система компонує остаточний комплекс Web-сервісів підтримки бізнес процесів на основі моделей оптимізації;

3. Автоматичний (перспективний) – на основі семантичних описів бізнес-процесу і Web-сервісів система здійснює пошук останніх, відбирає їх комплекс на основі моделей оптимізації і автоматично формує та виконує бізнес-процеси.

Такий підхід враховує реалії і вже сьогодні дозволяє підтримувати автоматизоване проектування і виконання бізнес-процесів в організації підприємства чи компанії. Ці можливості забезпечуються засобами базового і автоматизованого режимів роботи системи.

Важливим недоліком перших двох режимів є “жорстка” прив’язка Web-сервісів до бізнес-процесів. У випадку будь-яких змін структури чи повної відмови Web-сервісів, відповідний бізнес-процес не зможе бути правильно виконаний.

Щоб забезпечити певний рівень незалежності бізнес-процесів від Web-сервісів і потрібен третій автоматичний режим роботи. Однак, його експериментальне дослідження, здійснюване в університетських лабораторіях, ще не перейшло в фазу дослідного впровадження на реальних підприємствах.

Тут є декілька причин: по-перше, семантичні описи ще не досягли того рівня, щоб на їх основі однозначно автоматично визначити чи здатен Web-сервіс виконати елемент бізнес-процесу, по-друге, підготовка і перехід до такої схеми вимагають часу і витрат, що враховуючи першу причину не є наразі доцільним.

Зазначимо, що від обраного режиму роботи залежить лише метод пошуку можливих реалізацій завдань бізнес-процесу. Таким чином перехід від базового і автоматизованого режимів до більш перспективного автоматичного може

бути реалізований без суттєвого втручання до всієї системи.

Аналіз бізнес-процесу та генерація методів

Нехай бізнес процес, який необхідно автоматизувати, описаний мовою BPEL. Мова BPEL, використовуючи мову XML за основу, підтримує стек технологій розробки та надання Web-сервісів, в тому числі SOAP, WSDL, UDDI, WS-Reliable Messaging, WS-Addressing, WS-Coordination, WS-Transaction. Процес, описаний мовою BPEL, складається з кроків, які називаються “завданнями”. Мова BPEL підтримує примітивні, а також структуровані завдання.

Структуровані завдання, або підпроцеси, є невід’ємною складовою частиною мови BPEL. Вони дозволяють будувати бізнес-процес, використовуючи рівні абстракції. Таким чином, при правильній побудові, на кожному рівні бізнес-процес залишається простим і зрозумілим. Структуровані завдання повністю складаються з примітивних завдань.

Примітивні завдання є основними конструкціями і використовуються для загальних задач, однією з яких є виклик інших Web-сервісів. Таке завдання в XML представленні бізнес-процесу позначається тегом <invoke>. Тобто, для виявлення системних завдань як у складі процесу, так і у складі структурованих завдань достатньо віднайти такі теги у тексті XML. Приклад такої частини наведений нижче.

```
<invoke partnerLink="concretePartnerLink"
portType="pt:concretePortType"
operation="concreteOperation" input-
?variable="in_variable" output-
?Variable="out_variable" />
```

Для кожного знайденого системного завдання має бути згенерований відповідний метод Web-сервісу системи. Наприклад, для наведеного прикладу має бути згенерований метод “concreteOperation”. Після цього, за допомогою одного з режимів роботи, згенерованому методу ставиться у відповідність виклик зовнішнього Web-сервісу результат якого буде повернений згенерованим методом.

Пошук реалізацій завдань бізнес процесу в залежності від режиму роботи

Введемо ряд позначень.

Процес P складається з ряду завдань $T_k, k = 1, \dots, N_t$, де N_t – кількість завдань процесу.

$W_i, i = 1, \dots, N_w$ – i -ий сервіс в реєстрі Web-сервісів, де N_w – кількість Web-сервісів в реєстрі.

$M_{ij}, j = 1, \dots, N_{mi}$ – j -ий метод, що виконується i -им Web-сервісом, N_{mi} – число методів i -го Web-сервісу.

$M = \{M_{ij}\}$ – множина всіх методів Web-сервісів.

$impl(T_k, M_{ij}) \in \{0, 1\}$ – функція реалізації, що приймає значення 1, якщо метод M_{ij} реалізує завдання T_k процесу P , і 0 в супротивному випадку.

$X(T_k)$ – множина реалізацій завдання T_k , де

$$X(T_k) = \{M_{ij} | impl(T_k, M_{ij}) = 1\}$$

$x(P) \in \prod_{k=1}^n X(T_k) = \{M_{ij}^1, \dots, M_{ij}^k\}$ – одна з реалізацій процесу.

$X(P)$ – множина реалізацій процесу P , де

$$X(P) = \{x(P)\} = \prod_{k=1}^n X(T_k)$$

Таким чином, процес пошуку необхідних Web-сервісів можна описати наступним чином:

1. В базовому режимі оператор для кожного завдання бізнес-процесу T_k ставить у відповідність деякі методи M_{ij} , вручну формуючи множину реалізацій бізнес-процесу $X(P)$.

2. В автоматизованому режимі для кожної операції T_k оператору пропонується множина методів можливих реалізацій завдання $M' \in M, |M'| \ll |M|$, яка сформована методами синтаксичного пошуку серед доступних UDDI реєстрів. Тепер, для кожного завдання бізнес-процесу T_k оператор відбирає методи M_{ij} , такі що $impl(T_k, M_{ij}) = 1$, формуючи множину реалізацій завдання $X(T_k)$, з яких формується множина реалізацій процесу $X(P)$. При чому, для $\forall k X(T_k) \neq \emptyset$, інакше процес не може бути підтримано.

3. В автоматичному режимі для кожної операції T_k система автоматично знаходить множину реалізацій завдання $X(T_k)$ з якої формується множина реалізацій процесу $X(P)$. Це стає можливим за допомогою семантичних методів опису Web-сервісів. Проте, як вказувалось раніше, ці технології недостатньо розвинені і розповсюджені.

Синтаксичний пошук Web-сервісів.

Розглянемо Web-сервіс як програмну систему, що визначається рядком URI і інтерфейсами, визначеними на мові XML. Опис цієї програмної системи може бути знайдено іншими програмними системами, які можуть взаємодіяти з нею згідно з описом, за допомогою XML повідомлень. Шукати Web-сервіси можна вручну на сайтах виробників або сайтах-агрегаторах. Для машинного пошуку Web-сервісів використовують власні UDDI реєстри та реєстри бізнес-партнерів. Також для машинного пошуку можна використовувати відкриті UDDI реєстри, що підтримуються ентузіастами, а також їх альтернативи. Пошук Web-сервісів може бути виконаний по класифікаційному коду Web-сервісу, що присвоюються за класифікаторами (галузевого класифікатора NAICS, класифікатору продуктів і послуг UNSPSC, географічному класифікатору ISO 3166) при реєстрації Web-сервісу в реєстрі та/або ключовим словами, які приписуються Web-сервісу виробником. Тобто використовуючи набір ключових слів можна суттєво зменшити область пошуку необхідного Web-сервісу.

Таким чином, пошукова модель Web-сервісу являє собою двійку $\langle KK, KC \rangle$, де KK – це класифікаційний код сервісу, а KC – множина ключових слів, що описують Web-сервіс. Запити на пошук також будуються відповідно до цієї моделі: або задається класифікаційна рубрика, по якій знаходяться всі коди, що входять в цю рубрику, і відповідні їм сервіси; або задається набір ключових слів, за яким будується ешелонована видача сервісів, релевантних запиту; або задається і те й інше. У будь-якому випадку результати запиту вимагають осмислення з точки зору відповідності реальним вимогам. Тобто, система автоматизації має створювати запит на перевірку отриманих результатів.

Для звуження кола пошуку необхідно також перевірити на відповідність методи Web-сервісу. Припустимо, що Web-сервіс W має два набори параметрів: $W^i = \{I_1, I_2, \dots\}$ для SOAP запиту (вхідні параметри) та $W^o = \{O_1, O_2, \dots\}$ для SOAP відповіді (вихідні параметри). Коли W виконується з усіма вхідними параметрами, W^i , він повертає вихідні параметри, W^o . Припустимо, що для того, щоб викликати W , всі вхідні параметри в W^i повинні бути вказані (тоб-

то, W^i є обов'язковими параметрами). Позначимо також набір параметрів як P , а набір Web-сервісів як W .

Коли "значення" двох параметрів, $p_1 \in P$ та $p_2 \in P$, є взаємозамінними, загалом, ми можемо сказати, що вони знаходяться "у відповідності" один з одним. Найпростіший спосіб перевірки на відповідність коли два параметри мають однакове ім'я і тип: $(p_1.name = p_2.name) \cap (p_1.type = p_2.type)$.

Тобто, у найпростішому випадку, до області допустимих методів, потрапляють методи Web-сервісів, що підходять за ключовими словами і параметри цих методів відповідають необхідним.

Семантичний пошук Web-сервісів.

Основні існуючі стандарти Web-сервісів (SOAP, WSDL і UDDI) спроектовані для представлення інформації про інтерфейси сервісів, їх розгортання та виклики, але практично не представлені можливості сервісів, що обмежує автоматичну інтеграцію додатків, написаних у відповідності зі стандартами Web-сервісів.

Проблема автоматичної інтеграції Web-сервісів викликала появу підходу побудованого на семантичних описах Web-сервісів. Семантично описані Web-сервіси становлять собою Web-сервіси, чиї «властивості, можливості, інтерфейси і результати кодуються в однозначній формі, що піддаються машинному опрацюванню» [5]. У цьому випадку, Web-сервіси можуть бути проаналізовані, як з точки зору зіставлення знання мети та інтерфейсів, так і з точки зору семантичного опису сервісів для реалізації пошуку, композиції, запуску, моніторингу та ін.

На сьогодні, загально визнаний реєстр для семантичних описів Web-сервісів відсутній, хоча є пропозиції про включення такого опису в стандарт UDDI [6]. Передбачається, що семантичний опис сервісу буде виконуватися розробником відповідно до моделі $\langle I, O, P, E \rangle$. Це означає, насамперед, що описуються не сервіси, а пов'язані з ними абстрактні процеси, що плануються до виконання в конкретній предметній області. Зазначена область задається концептуальною моделлю у вигляді OWL-онтології. I, O у цій моделі – це множина типів даних, що задаються класами онтології на вході і виході процесу відповідно; P – це множини логічних виразів, що визначають передумови, при виконанні яких можливе успішне виконання проце-

су; E – множина логічних виразів, що відображають ефекти виконання процесу з точки зору зміни стану зовнішнього середовища.

Очевидно, що і запит на пошук процесів повинен будуватися відповідно до моделі $\langle I, O, P, E \rangle$ – по суті, користувач, складаючи запит, повинен сформулювати семантичний опис потрібного процесу. Це можливо, якщо онтології, що використовуються для формування запиту і процесу, ідентичні або зводяться одна до іншої.

Пошук процесів на основі відповідності їх входів/виходів запитом теоретично становить тільки проміжний етап у вирішенні проблеми пошуку сервісів за їх семантичним описам. Багато дослідників цієї проблеми зупинилися на цьому етапі, оскільки не могли знайти місце передумовам і ефектам в алгоритмах пошуку. Тому, переважна більшість досліджень, пов'язаних з пошуком процесів семантичних Web-сервісів, будується на основі моделі запиту у вигляді $\langle I, O \rangle$. Оскільки I, O – це множина типів даних на вході і виході процесу, що задаються класами онтології предметної області, то вноситься семантична складова в пошук процесів. Цим пошук процесів семантичних Web-сервісів відрізняється від пошуку Web-сервісів за ключовими словами.

Вибір оптимальної реалізації серед множини можливих реалізацій

Після пошуку реалізацій завдань бізнес-процесу маємо сформовану множину реалізацій бізнес-процесу $X(P)$. Для того, щоб обрати єдину оптимальну реалізацію $x(P) \in X(P)$ необхідно визначити критерії оптимальності.

У статті для оцінки методів Web-сервісів M_{ij} використовують такі функції:

$f_1(M_{ij}) = \text{ExecutionTime}(M_{ij})$ – час виконання і обробки методу M_{ij} в секундах;

$f_2(M_{ij}) = \text{Cost}(M_{ij})$ – вартість 1000 запитів до Web-сервісу W_i у гривнях;

$f_3(M_{ij}) = \text{FreeLimit}(M_{ij})$ – кількість безкоштовних запитів до Web-сервісу W_i на день.

Використовуючи, функції оцінки методів, визначимо відповідні функції оцінки реалізацій процесів $x(P)$:

$$f_1(x(P)) = \sum_k f_1(M_{ij}^k),$$

$$f_2(x(P)) = \sum_k f_2(M_{ij}^k),$$

$$f_3(x(P)) = \sum_k f_3(M_{ij}^k), k = 1, \dots, N_t, \partial e$$

M_{ij}^k – метод M_{ij} , що реалізує k -те завдання бізнес-процесу P .

Введемо також функцію оцінки реалізації бізнес-процесу $x(P)$, що не визначається суммою функцій оцінок відповідних методів:

$f_4(x(P)) = \text{ServiceCount}(x(P))$ – кількість унікальних Web-сервісів задіяних в реалізації $x(P)$

Сформулюємо проміжну задачу наступним чином: серед множини можливих реалізацій процесу $X(P)$ вибрати таку реалізацію $x(P) \in X(P)$ для яких

$$\begin{aligned} f_1(x) &= f_1(x(P)) \rightarrow \min, \\ f_2(x) &= f_2(x(P)) \rightarrow \min, \\ f_3(x) &= f_3(x(P)) \rightarrow \max, \\ f_4(x) &= f_4(x(P)) \rightarrow \min. \end{aligned}$$

Спростимо цільову функцію, ввівши векторний критерій, який приймає значення у просторі m -мірних векторів R^m , та звівши все до задачі мінімізації:

$$f(x) = f(x(P)) \rightarrow \min,$$

$$\begin{aligned} f(x) &= (f_1(x), f_2(x), -1 * f_3(x), f_4(x)), \\ f(x) &\in R^m, x \in X(P) \end{aligned}$$

Для більш точного визначення оптимальної реалізації згідно нагальних потреб у системі використовується поняття відносної важливості критеріїв, за яким деякий i -ий критерій f_i важливіший за j -ий критерій f_j з заданими додатними параметрами w_i, w_j . Оскільки важливість може змінюватися з часом, інформація про відносну важливість критеріїв винесена в конфігураційний файл. Якщо відносна важливість не вказана використовується відношення важливості за замовчуванням, за яким кількість сервісів, що використовується важливіша за час виконання, час виконання важливіший за ціну запиту, яка в свою чергу важливіша за ліміт безкоштовних запитів на добу:

$$\Omega = \{\text{ServiceCount} > \text{ExecutionTime}, \text{ExecutionTime} > \text{Cost}, \text{Cost} > \text{FreeLimit}\} =$$

$= \{f_4(x) > f_1(x), f_1(x) > f_2(x), f_2(x) > f_3(x)\}$
Остаточна задача оптимізації виглядає наступним чином:

$$\begin{cases} f(x) \rightarrow \min \\ x \in X(P) \\ \Omega \end{cases}$$

Для вирішення задачі в системі використовується метод цільового програмування з евклідовою метрикою.

Приведемо алгоритм вирішення задачі:

1. Провести оцінку множини можливих реалізацій $X(P)$;
2. Сформувати множини Парето-оптимальних реалізацій [7];
3. Звузити множини Парето за рахунок інформації про важливість критеріїв [8];
4. Вирішити задачу цільового програмування $x^* \in X^*(P)$ [9]:

$$\text{inf}_y \rho(f(x^*), y) = \min_{x \in X} \text{inf}_y \rho(f(x), y), \text{ де}$$

ρ – евклідова відстань;

$X^*(P)$ – звужена множина Парето;

$y = (0, 0, -1 * \max_{x \in X} f_3(x), 0)$ – ідеальний вектор критеріїв оцінки.

Висновки

Запропонована система автоматизації підтримки бізнес-процесів, яка розв'язує ряд важливих завдань, насамперед, автоматизації пошуку можливих реалізацій завдань бізнес-процесу та вибору оптимальної реалізації серед усіх можливих. При цьому надається єдиний інтерфейс для взаємодії між методами бізнес-процесу та їх реалізаціями. Система, побудована таким чином, робить основний бізнес-процес майже нечутливим до зовнішніх змін. Крім того, система побудована з урахуванням можливого переходу до третього автоматичного режиму, в якому множина реалізацій бізнес-процесу буде визначатися повністю машиною за рахунок семантичного пошуку, що дозволить однозначно визначати необхідні Web-сервіси та позбавить від необхідності втручання людини.

Список посилань

1. Information technology – Object Management Group Business Process Model and Notation, ISO/IEC 19510, 2013;

2. Мохаммед И. Мабрук. Краткие основы SOA // IBM Developer Works [Електронний ресурс]. – 2010. – Режим доступу до ресурсу: <http://www.ibm.com/developerworks/ru/edu/ws-soa-ibmcertified/index.html>.
3. Web Services Architecture – W3C Working Group Note 11. – The World Wide Web Consortium (W3C) [Електронний ресурс]. – 2004. – Режим доступу до ресурсу: <http://www.w3.org/TR/ws-arch/>.
4. ORACLE Data Sheet – ORACLE BPEL Process Manager – Режим доступу: <http://www.oracle.com/technetwork/middleware/bpel/overview/ds-bpel-11gr1-1-134826.pdf>;
5. McIlraith S. Semantic Web services / McIlraith S., Son T.C., Zeng H. // IEEE Intelligent Systems. Special Issue on the Semantic Web. – 2001. – Vol. 16. – N. 2. – P. 46-53.
6. Климов В.В. Система поиска и интеграции веб-сервисов, основанная на семантических описаниях / В.В. Климов // Приоритетные направления инновационного развития. – 2009. – №1. – С. 33.
7. Подиновский В. В. Парето-оптимальные решения многокритериальных задач / В. В. Подиновский, В. Д. Ногин. – М: Наука. Гл. ред. физ.-мат. лит.-ры, 1982. – 256 с..
8. Ногин В. Д. Принятие решений в многокритериальной среде: количественный подход (2-е изд.) / В. Д. Ногин. – М: Физматлит, 2005. – 176 с.
9. Поляк Б. Т. Введение в оптимизацию / Б. Т. Поляк. – М: Наука. Гл. ред. физ.-мат. лит.-ры, 1983. – 384 с.