

**5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

УДК 004.021

МЕТОД ВИБОРУ ВЕБ-СЕРВІСУ НА ОСНОВІ ПЕРЕТИНУ КРИТЕРІАЛЬНОЇ З ПАРЕТО-ОПТИМАЛЬНОЮ МНОЖИНОЮ

Хмелюк М.С., Захарчук Б.О В

Національний технічний університет України «Київський політехнічний інститут», м.Київ
E-mail: bogdinho@ukr.net

ORCID: 0000-0003-1516-6153

Copyright © 2014 by author and the journal “Automation technological and business - processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>ONAFT
Open Access

DOI: 10.15673/2312-3125. 21/

Анотація

В даній статті розглянуто переваги і недоліки сервіс-орієнтованої архітектури, а також її застосування на конкретному практичному використанні. Проаналізовані існуючі методи вибору веб-сервісів із множини зареєстрованих сервісів, шляхом знаходження оптимуму Парето або опираючись на вектор критеріїв прикладений до множини доступних сервісів. Також в статті наведено власний розроблений метод пошуку оптимального веб-сервісу з множини зареєстрованих, наведено його найбільші переваги та недоліки, над якими ще потрібно працювати. Наведені результати порівняння розробленого методу з розглянутими існуючими методами. В результаті отримано метод, який можна інтегрувати в будь-яку сервіс-орієнтовану архітектуру, нову чи оптимізувати роботу існуючої.

Annotation

In this article the advantages and disadvantages of a service-oriented architecture, and its application in a particular practical use. Existing methods for selecting Web services plurality of registered services by finding Pareto optimum or relying on vector set of criteria applied to the available services. Also in the article are his own developed method of finding the optimal service of a plurality of registered, are its biggest advantages and disadvantages over which still need work. The results compare the proposed method with existing methods discussed. As a result, received a method that can be integrated in any service-oriented architecture, new or optimize existing.

Ключові слова

Програмне забезпечення, сервіс-орієнтована архітектура, синтез бізнес-процесів, веб-сервіс, множина Парето, композиція веб-сервісів.

Вступ

Більшість складних сучасних програмних комплексів пов'язані з веб-сервісами, які виконуються на віддалених серверах, завдяки сервіс-орієнтованій архітектурі, обчислення деяких операцій та виконання окремих функцій може бути покладено на іншу обчислювану машину, яка доступна через мережу інтернет. До цих пір були запропоновані різні методи планування і вибору веб-сервісів, в яких розглядається проблема автоматизованої композиції веб-сервісів. Однак, в реальних випадках, проблема планування ще далека від тривіальної:



5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

планувальник повинен мати справу з недетермінованою поведінкою веб-сервісів, частковою спостережністю їх внутрішнього стану, а також з комплексними цілями, наприклад, вираження часових. Використання веб-сервісів у програмних комплексах надає наступні переваги:

- Використання існуючих сервісів для виконання операцій, які можуть бути використані нашим програмним комплексом;
- Стає простішою інтеграція функціоналу нашого програмного комплексу з іншим комплексом, спорідненим за бізнес-процесами;
- Можливість масштабування програмного комплексу, збільшення навантаження на обчислювальні операції, шляхом встановлення додаткових веб-серверів з сервісами, які використовує програмний комплекс.

Існують окремі класи задач, які потребують сервіс-орієнтованого підходу. Можна навести приклад простої задачі: формування звіту по прибуткам за місяць, тобто для виконання цієї задачі може бути задіяний сервіс, який викликає процедуру на сервері бази даних.

Також за допомогою сервіс-орієнтованого підходу можна вирішувати набагато складніші задачі, які потребують втручання алгоритмів по вибору того чи іншого сервісу для виконання поставленої задачі. Наприклад, потрібно знайти найближчу аптеку, де продаються конкретні ліки. В цьому випадку може бути задіяно декілька сервісів, їх вибір та порядок виконання лягає на алгоритм, який я спробую представити у даній статті.

Сформулюємо основні переваги та спектр задач, які може вирішувати програмний комплекс з сервіс-орієнтованою архітектурою:

- Збільшення функціональності програмного комплексу за рахунок використання існуючих сервісів;
- Підвищення продуктивності систем, за рахунок використання віддалених обчислювальних серверів;
- Збільшення надійності та відмовостійкості програмного комплексу;
- Надання можливості запровадження власного методу вибору сервісів, для виконання задач;
- Підвищення якості інформації системи.

Збільшення функціональності програмного комплексу мається на увазі додавання посилань на існуючі сервіси, які виконують аналогічні з нашим програмним комплексом операції, шляхом мінімального втручання у вихідний код.

Підвищення продуктивності досягається шляхом розподілення виконання операцій на різні сервіси або при умові великого навантаження на один і той самий сервіс, підключення ще одного сервісу на іншому віддаленому сервері, щоб забезпечити паралельне виконання роботи.

Збільшення надійності та відмовостійкості програмного комплексу досягається наявністю в системі альтернативних додаткових серверів з встановленими додатковими сервісами, які в будь-який момент часу можуть бути задіяні.

Завдяки можливості запровадження власного методу користувачі системи матимуть змогу оптимізувати роботу того чи іншого бізнес-процесу, наприклад побудови черги виклику чітко вказаних сервісів, чи надання якоїсь множини сервісів, які можуть вирішувати конкретну задачу.

Підвищення якості інформації може бути досягнуто шляхом порівняння і аналізу результатів виконання операцій аналогічними сервісами, також потрібно враховувати їхню швидкодію. Якщо швидкодія значно відрізняється потрібно шукати шляхи для заміни повільного сервісу якоюсь альтернативою.

Недоліки такого підходу можуть виникнути при:

- Використанні методу у банківській сфері або у сфері, яка тісно пов'язана з грошовими транзакціями, оскільки система може бути налаштована на кількох віддалених серверах, з'являється додатковий незахищений від атак рівень;
- Адмініструванні, оскільки налаштування і супроводження віддалених сервісів може виникнути проблема з мережевими обладнаннями, файрволами, логічними змінами у мережі, як наслідок сервіс буде не доступний.

Розглянемо основні підходи для побудови сервіс-орієнтованих систем з алгоритмом вибору коректних сервісів для розв'язання поставленої задачі.

Аналіз існуючих рішень

В роботах [1-3] розглядається метод вибору сервісів на основі обчисленої репутації, тобто вводиться така міра оцінки сервісів, як «репутація», на основі якої метод обирає сервіс для розв'язання тієї чи іншої задачі. Цей підхід удосконалив вибір сервісу у сервіс-орієнтованих системах з урахуванням репутації на основі багатокритеріальної оптимізації, яка відрізняється використанням нелінійної схеми компромісів, що забезпечує Парето-оптимальне

5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

рішення, див. Рис.1. Також в цій роботі описане удосконалення моделі функції корисності ґсновану на аксіоматиці Неймана-Моргенштерна, що на відміну від існуючих враховує додаткові параметри, що підвищує стійкість моделі до характерних сценаріїв компрометації репутації. Також потрібно відзначити недоліки цього підходу.

Недоліки існуючих моделей репутації:

- Відсутній математичний формалізм, для опису моделей використовується мова псевдокоду;
- Моделі будуються на основі суб'єктивних оцінок;
- Репутація не враховується для автоматичного розподілу задач по сервісам;
- Призначена для статичного планування (наперед задана кількість задач);
- Спочатку всі ресурси вважаються надійними;
- Не стійка до сценаріїв компрометації репутації.



Рис.1 - Діаграма послідовності для алгоритму з критерієм «репутація» сервісів

Критерій «репутація» гарна ідея для аналізу і вибору сервісів для вирішення поставленої задачі, але немає чіткого алгоритму для обчислення цієї репутації, тобто потрібно удосконалити алгоритм обчислення репутації та забезпечити постійне оновлення цього критерію на основі всіх проведених операцій пов'язаних з конкретним сервісом.

В роботах [4-6] розглядається метод планування для автоматизованої композиції веб-сервісів, які описані в OWL-S моделях процесу, який може ефективно боротися з недетермінізмом, частковою спостережністю і складними цілями. Метод дозволяє синтез планів, які кодують композиції веб-сервісів звичайними конструкціями програмування, наприклад, умовні оператори й ітерації. Створені таким чином плани можуть бути трансльовані у виконувані процеси.

Під автоматизованою композицією ми маємо на увазі задачу автоматичної генерації, враховуючи набір доступних веб-сервісів, нового веб-сервісу, який досягає поставленої мети, взаємодіючи з (деякими) доступними веб-сервісами. Більш конкретно, ми взяли за відправну точку OWL-S онтологію моделі процесу [8], тобто декларативний опис програми, яка реалізує сервіс. Враховуючи OWL-S опис моделі процесу П доступних сервісів (W_1, \dots, W_n), ми кодуємо кожного з них у систему переходу з одного стану в інший ($\Sigma W_1, \dots, \Sigma W_n$), див. Рис 2. Системи переходів забезпечують свого роду операційну семантику до OWL-S моделей процесу. Кожна з них описує відповідний веб-сервіс як динамічну систему на основі стану, яка може розвиватися, наприклад, змінювати стан, і яка може бути частково контрольованою і спостерігатися зовнішніми агентами.

G - шукана композиція веб-сервісів; $W_{1..n}$ - зареєстровані веб-сервіси; Σ - залучення алгоритму пошуку оптимальних веб-сервісів;

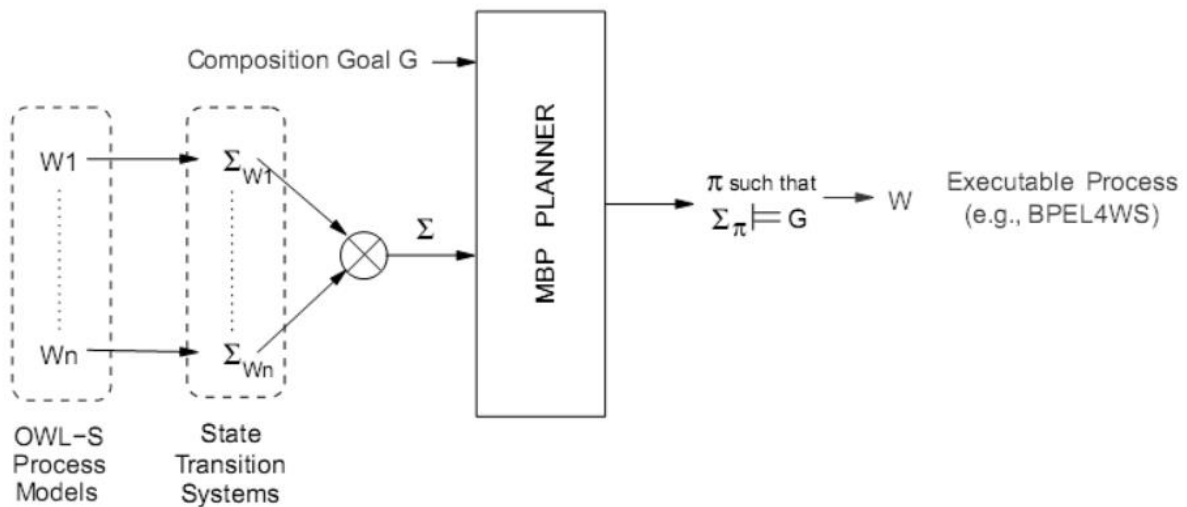
5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

Рис.2 - Автоматизована композиція на основі OWL-S

З точки зору нового композитного сервісу, який повинен бути створений, скажімо W , системи переходів $\Sigma W_1, \dots, \Sigma W_n$ являють собою середовище, в якому W повинен працювати по прийому й відправленню запитів на сервіс. Вони складають те, що в літературі по плануванню, називається доменом планування, тобто область, де планувальник планує мету. У нашому випадку, домен планування являє собою систему переходів, який поєднує $\Sigma W_1, \dots, \Sigma W_n$. Формально, ця комбінація є синхронним продуктом, який дозволяє п сервісам розвиватися незалежно і паралельно, тому всі можливі поведінки, еволюції області планування, без будь-якого контролю, виконаного сервісом, який буде генерований, тобто, W .

З недоліків цього підходу потрібно відмітити, що динамічна система може сама розвиватися, наприклад змінювати стан і в той самий момент вона частково контрольована і може спостерігатися зовнішніми агентами, що є неприпустимо з точки зору безпеки. Це може призвести до зовнішнього втручання сторонніх сервісів у виконання певних операцій, а якщо сервіси працюють з базами даних – може призвести до втрати цінних бізнес-даних.

Для усунення недоліку з безпекою використання сервісів можна ввести поняття реєстрації сервісів на сервері і перевіряти унікальний номер сервісу при виклику. Щодо часткового контролю динамічної системи, потрібно обмежити систему у розвитку надавши їй доступ тільки до сукупності сервісів, яка безпосередньо пов'язана з конкретно виконуваною функцією.

В роботі [9] розглядається модель композиції сервісів, яка базується на антологіях, що базується на декларативних описах сервісів. Формальною основою для композиції є синтаксичні і семантичні характеристики сервісів, які визначають є сервіс композитним чи ні. Для вибору сервісу для виконання операції використовуються якісні показники: вартість, безпека і секретність, також можливі введення додаткових критеріїв.

Варто відмітити переваги даного методу з точки зору безпеки. Сервіси при реєстрації на сервері можна помічати як секретний, який буде використовуватись тільки, наприклад, фінансовими операціями чи державної важливості операціями.

З недоліків те, що немає чітко описаного алгоритму оцінювання сервісу по критеріям, також немає пріоритетності по критеріям у випадку однакових показників. Не передбачено алгоритм планування виконання сервісів.

Якщо узагальнити всі розглянуті методи вибору сервісів, то можна прийти до висновку, що алгоритм і його складність у виборі сервісів залежить від програмного комплексу, де він буде застосований. Його доцільність потрібно аналізувати окремо, адже це впливає на швидкодію інфраструктурного середовища і складність його встановлення, налаштування і супроводження.

Результати аналізу представлені в таблиці 1.

**5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

Система Якість	Метод репуатації сервісів	Метод планування автоматизованої композиції	Метод декларативного опису сервісу, вибір по критеріям
Масштабування	+	+	+
Централізоване використання	+	+	+
Децентралізоване використання	+	+	-
Можливість розподіленого виконання	-	+	+
Можливість відкладеного виконання	-	+	-
Можливість аналізу сервісів і вибір у ручному режимі	-	-	+
Можливість налаштування додаткових критерії оцінки	-	-	+
Можливість секретного виконання, перевірка секретності	-	+	+

Запропоноване рішення

Задачу вибору сервісу для виконання конкретної задачі можна розділити на дві під задачі: описання, реєстрація сервісів на сервері й алгоритм вибору сервісів з множини зареєстрованих сервісів. Розв'язання окремої задачі не може існувати без розв'язання іншої. Для початку розглянемо задачу у вигляді математичної моделі.

В моделюванні задачі використовується нотація теорії важливості критеріїв. Нотація сервісів і функцій має надстроковий індекс, нотація оцінок значень критеріїв - під строковий індекс

- Задача T складається з ряду під задач: t^i , $i = |1, N_t|$, де N_t — число під задач задачі T .

- S^i , $i = |1, N_s|$ — i -ий сервіс в реєстрі веб-сервісів, де N_s — число веб-сервісів в реєстрі.

- S^i, f^j , $j = |1, N_{fi}|$ — j -а функція, яка реалізується i -им веб-сервісом, N_{fi} — число функцій i -го сервісу.

- $Implement(t^k) = \{S^i, f^j \mid S^i, f^j := t^k\}$ — множина можливих реалізацій під задач t^k за допомогою функцій з сервісів S^i, f^j .

- $Implement(T) = \{Implement(t^k) \mid k = |1, N_t|\}$ — множина можливих реалізацій задачі T , або можливі композиції сервісів в рамках задачі T .

- $K(S^i, f^j) = (K_1(S^i, f^j), \dots, K_{N_k}(S^i, f^j))$ — векторна оцінка j -ої функції i -го веб-сервісу по K_n , $n = |1, N_k|$ критеріям.

Розглянемо кожну з під задач більш детально та запропонуємо підхід до їхнього розв'язання. Графічне представлення математичної моделі показано на Рис.3.

S - Доступні сервіси; f - функції, які надають сервіси; K - критерії, за якими відбираються сервіси;

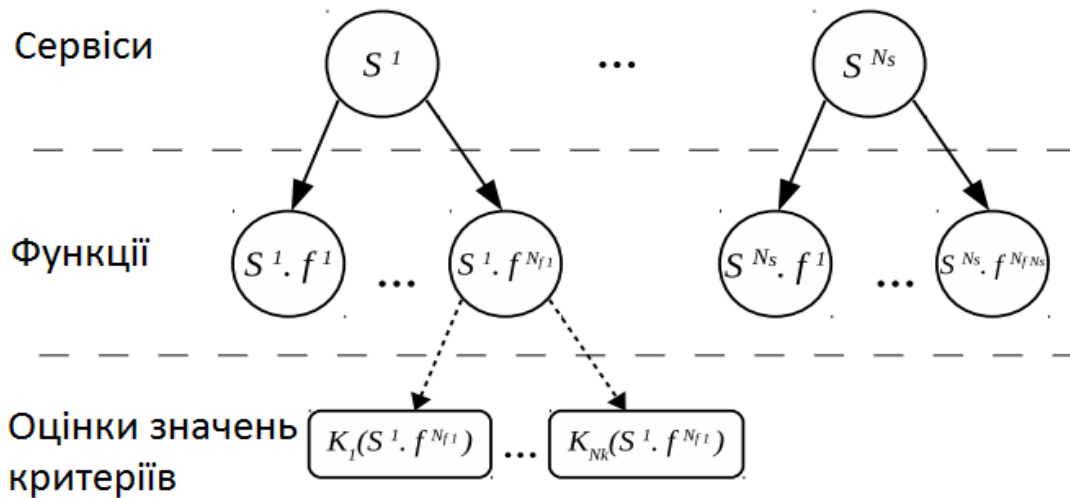
5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

Рис. 3 – Математична модель описання сервісів

Для описання сервісів будемо використовувати XML-подібну мову – WSDL. Сервіси доступні через операції, кожна з яких ідентифікується іменем і текстовим описанням, яке узагальнює її функціональні особливості. Операція також визначає спосіб передачі повідомлень, вхідні та вихідні повідомлення, ціль і категорія. Далі представимо визначення операцій сервісу.

Операція сервісу OP_{ik} визначена кортежем

($Description_{ik}$, $Mode_{ik}$, In_{ik} , Out_{ik} , $Purpose_{ik}$, $Quality_{ik}$, $Responsibility_{ik}$, $Priority_{ik}$, $Secret_{ik}$), де

- $Description_{ik}$ – текстове описання функціональної операції;
- $Mode_{ik}$ – тип операції (*one_way*, *notification*, *request-response*);
- In_{ik} і Out_{ik} – вхідне чи вихідне повідомлення;
- $Purpose_{ik}$ – описує бізнес-функцію операції;
- $Quality_{ik}$ – визначає якісні характеристики операції;
- $Responsibility_{ik}$ – визначає розділ, який покриває операція;
- $Priority_{ik}$ – визначає пріоритетність операції;
- $Secret_{ik}$ – визначає, чи потрібно перевіряти зареєстрований сервіс у валідності.

Для використання методу вибору сервісів з сукупності зареєстрованих будемо користуватися словником критеріїв порівняння сервісів наведених на Рис.4.

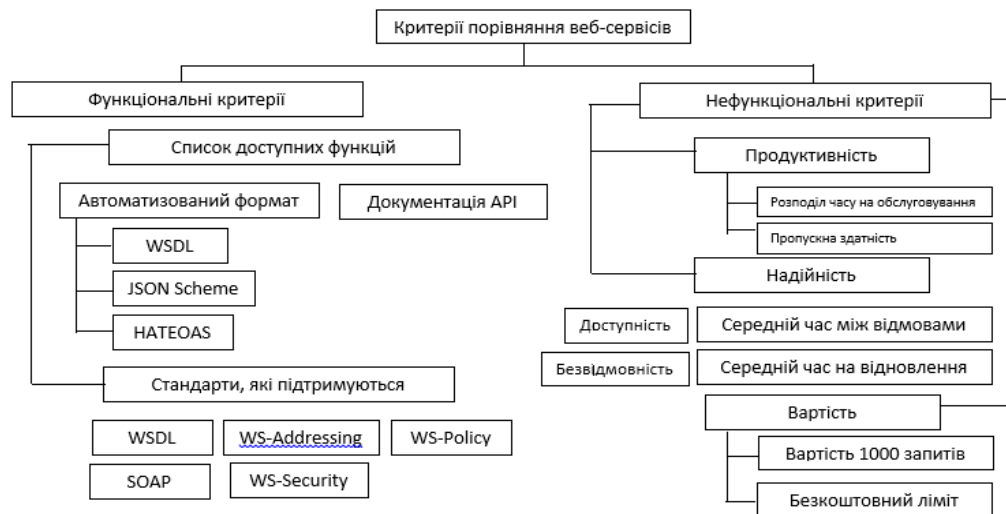


Рис. 4 – Критерії порівняння сервісів

5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ

Тобто рішення задачі може бути узагальнене і буде можливість гнучкого налаштування для конкретного бізнесу, цю опцію можна реалізувати за допомогою вбудованого довідника з критеріями і налаштувати його на початку роботи з програмним комплексом.

Довідник з критеріями мається на увазі стандартизовані одиниці виміру сервісів, які можуть бути застосовані для налаштування алгоритму вибору сервісів при виконанні операцій бізнес-процесу. Також буде доступна функція створення індивідуальних одиниць виміру. Налаштування довідника буде проходити на етапі інсталяції системи.

Наступний етап це реєстрація доступних сервісів, які будуть використані для здійснення усіх операцій по кожному бізнес-процесу. Для описання сервісів будемо використовувати запропоновану специфікацію OP_{ik} .

Отже, задача по вибору бажаного сервісу зводиться до знаходження множини сервісів, які задовольняють умовам критеріїв та потрапили до Парето-оптимальної множини. Даний метод називається, **перетин критеріальної і Парето-оптимальної множини**. Суть методу вибору сервісів для виконання операції конкретного бізнес-процесу полягає у звуженні множини можливих у використанні сервісів шляхом аналізу таких властивостей, як: Responsibility area, Category, Purpose, Priority, Secret та інших з комбінуванням теорії формування множини Парето. Тобто це дає можливість в деякій мірі уникнути похибок розрахунків вибору сервісу з множини Парето. В такому випадку на вхід функції формування множини Парето будуть надходити тільки ті сервіси, які безпосередньо відповідають умовам операції бізнес-процесу.

Графічне зображення методу **перетину критеріальної і Парето-оптимальної множини** на Рис. 5.

S^p – possible services (множина можливих сервісів); C – набір критеріїв прикладений до вибору сервісу; S^{PO} – Парето-оптимальна множина; S – бажаний сервіс.

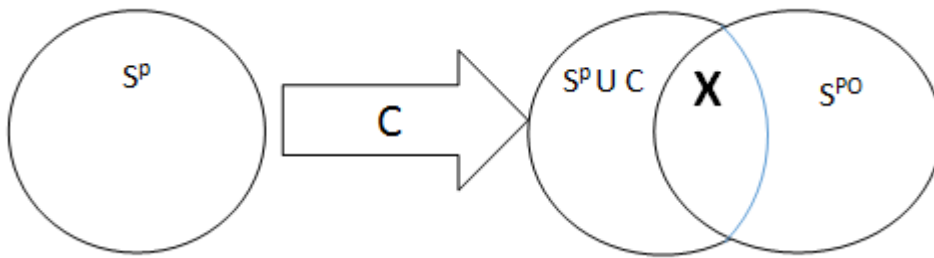


Рис.5 – графічне зображення методу перетину критеріальної і Парето-оптимальної множини.

Метод **перетину критеріальної і Парето-оптимальної множини** має наступні переваги:

- Можливість гнучкої конфігурації на етапі інсталяції, за рахунок функції словника;
- Прозора реєстрація сервісів, з опцією вказання секретності та пріоритетності;
- Вирішена проблема з можливістю втручання зловмисного коду на етапі обробки інформації на віддаленому сервісі, шляхом присвоєнню закодованого унікального ідентифікаційного номеру сервісу, який використовується віддалено;
 - Висока швидкодія, за рахунок попередніх обчислень множини можливих сервісів для кожного бізнес-процесу;
 - Усунута відома проблема низької швидкодії алгоритму Парето, за рахунок відсіювання явно зайвих варіантів на попередньому критеріальному етапі;
 - Можливість масштабування програмного комплексу;

Висновки

Аналіз існуючих підходів показав, що при вирішенні задачі вибору сервісу в сервіс-орієнтованих системах нарівні із врахуванням технічних та економічних аспектів виникає потреба у врахуванні моделей довіри на основі репутації. В якості покращення цього методу було запропоновано ймовірнісний опис моделі довіри на основі репутації, що дозволило формально обґрунтувати властивість стійкості моделі до характерних сценаріїв компрометації, також потрібно розробити окрему модель для оновлення репутації для кожного сервісу на основі усіх операцій виконаних конкретним сервісом. Також був розглянутий метод планування для автоматизованої композиції веб-сервісів, який може ефективно боротися з недетермінізмом, частковою спостережністю і складними цілями. Як виявилось в цьому методі є недолік пов'язаний з безпекою виклику сервісів і частковим контрольованою динамічною системою, відразу були запропоновані шляхи до усунення цих недоліків.

**5 ТЕОРІЯ І ПРАКТИКА АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ**

Запропонований власний комбінований метод для вибору сервісів з множини зареєстрованих сервісів, який дає можливість здійснити попередні розрахунки множини можливих сервісів, що підвищить швидкодію, а також зменшить похибку від обчислень остаточної множини сервісів формуючи множину Парето.

Література

1. Куссуль О.М. Исследование эффективности применения моделей доверия на основе репутации в Grid-системах / О.М. Куссуль, А.Н. Новиков, С.С. Швец // Наукові праці ДонНТУ Серія "Інформатика, кібернетика та обчислювальна техніка". – 2010. – Випуск 12(165). – С. 126-134.;
2. Куссуль О.М., Новиков А.Н. Многокритериальная оптимизация планирования выполнения задач в структурно-сложных системах на основе моделей репутации // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. — 2011. — № 54. — С. 36-47;
3. Куссуль О.М., Інформаційна технологія вибору сервісу в сервіс-орієнтованих системах з врахуванням репутації Вісник НТУУ «КПІ». Зб. наук. пр. — 2011. — № 54.;
4. S. Narayanan and S. McIlraith. Simulation, Verification and Automated Composition of Web Services. In Proceedings of the Eleventh International World Wide Web Conference (WWW-11), 2002.;
5. M. Pistore, P. Bertoli, F. Barbon, D. Shaparau, and P. Traverso. Planning and Monitoring Web Service Composition. In ICAPS'04 Workshop on Planning and Scheduling for Web and Grid Services, 2004.;
6. M. Sheshagiri, M. desJardins, and T. Finin. A Planner for Composing Services Described in DAML-S. In Proc. of Workshop on Web Services and Agent-based Engineering - AAMAS'03, 2003.;
7. Переклад статті «Automated Composition of Semantic Web Services into Executable Processes» P. Traverso and M. Pistore, Ремарович С.В.;
8. The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. In Technical White paper (OWL-S version 1.0), 2003.;
9. В. Дерещкий Подходы и задачи композиции сервисов в семантическом веб окружении. З наукового видання «Формальні методи розробки програмного забезпечення». УДК681.3.

References

1. Kussul Investigation of the effectiveness of application models based on reputation in Grid-systems / OM Kussul, AN Novikov, S.C. Shvets // Proceedings of DonNTU series "Informatics, Cybernetics and Computer Science". - 2010. - Issue 12 (165). - P. 126-134.;
2. Kussul AN, Novikov AN Scheduling optimization multicriterial systems perform tasks in structural complex systems based on models reputation // Vestnik NTU "KPI". Informatics, Management and Computer Science: Coll. Science. pr. - 2011. - № 54. - С. 36-47;
3. Kussul AM, Information technology selection service in service-oriented systems, taking into account the reputation of the Bulletin of NTUU "KPI". Coll. Science. pr. - 2011. - № 54.;
4. S. Narayanan and S. McIlraith. Simulation, Verification and Automated Composition of Web Services. In Proceedings of the Eleventh International World Wide Web Conference (WWW-11), 2002.;
5. M. Pistore, P. Bertoli, F. Barbon, D. Shaparau, and P. Traverso. Planning and Monitoring Web Service Composition. In ICAPS'04 Workshop on Planning and Scheduling for Web and Grid Services, 2004.;
6. M. Sheshagiri, M. desJardins, and T. Finin. A Planner for Composing Services Described in DAML-S. In Proc. of Workshop on Web Services and Agent-based Engineering - AAMAS'03, 2003.;
7. Translation of article 7. «Automated Composition of Semantic Web Services into Executable Processes» P. Traverso and M. Pistore, Remarovych SV.;
8. The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. In Technical White paper (OWL-S version 1.0), 2003.;
9. V. Deretsky Approaches tasks and compositions of services in semantic web environment. From the scientific journal "Formal methods of software development." UDK681.3