

УДК 004.056 (083.94)

DOI <https://doi.org/10.36994/2788-5518-2023-02-06-09>

ПИТАННЯ ВИКОРИСТАННЯ РЕКОМЕНДАЦІЙНИХ СИСТЕМ У СКЛАДІ БРОКЕРА ОБ'ЄКТНИХ ЗАПИТАНЬ В АРХІТЕКТУРІ СЕРВІС-ОРІЄНТОВАНОЇ СИСТЕМИ

Рогоза В. С., д-р техн. наук, професор, Західнопоморський технологічний університет, м. Щецин, Польща; Навчально-науковий комплекс «Інститут прикладного системного аналізу – ННК ІПСА», Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», м. Київ, Україна
<https://orcid.org/0000-0003-2327-156X>, wrogoza@wi.zut.edu.pl

Зарічний Я. С., Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», м. Київ, Україна
<https://orcid.org/0000-0002-7596-5857>, yarik120700@gmail.com

Анотація. У статті розглядається питання підвищення ефективності реалізації послуг у сервіс-но-орієнтованій архітектурі (SOA) шляхом інтеграції рекомендаційних систем до складу брокера об'єктних запитань. Автори аналізують ключові проблеми, що виникають під час функціонування двох компонентів SOA: шаблону виявлення послуг на боці сервера та шаблону використання реєстру послуг.

Для шаблону виявлення послуг визначено проблеми складності налаштування балансувальника, необхідності індивідуального балансування трафіку через гетерогенність сервісів, труднощі балансування навантаження на брокер, динамічності процесів обробки запитів, а також складності виконання моніторингу й аналізу запитів для виявлення аномалій і прогнозування навантаження.

Для шаблону використання реєстру послуг виявлено проблеми неструктурованості даних про сервіси, відсутності метаданих про їхні функціональні можливості, інтерфейси, доступність та якість, а також обмеження механізмів фільтрації сервісів, що ускладнює їх пошук і вибір.

Як рішення пропонується включити рекомендаційну систему до балансувальника навантаження та сервісу реєстрацій. Для балансувальника вона забезпечить оптимізацію розподілу трафіку, автоматичне виявлення аномалій, прогнозування навантаження, оптимізацію використання ресурсів та адаптивне управління процесами обробки запитів.

Для сервісу реєстрацій рекомендаційна підсистема дасть змогу динамічно оновлювати рекомендації відповідно до попиту, групувати користувачів за інтересами, надавати контекстно-чутливі рекомендації, збагачувати профілі користувачів, використовувати колаборативну фільтрацію та оцінювати якість рекомендацій на основі зворотного зв'язку.

Інтеграція рекомендаційних систем може значно підвищити ефективність, продуктивність і досвід користувачів SOA завдяки персоналізованим рекомендаціям релевантних сервісів, що відповідають їхнім конкретним потребам. Автори вважають це важливим кроком у розвитку сервіс-но-орієнтованої архітектури.

Ключові слова: сервіс-орієнтована архітектура, рекомендаційна система, балансувальник навантаження, реєстр сервісів, персоналізація, адаптивне управління.

USING RECOMMENDER SYSTEMS AS PART OF A BROKER OF OBJECT QUESTIONS IN THE ARCHITECTURE OF A SERVICE-ORIENTED SYSTEM

Valeriy Rogoza, dr hab., prof., West Pomeranian University of Technology, Szczecin, Poland; Educational and Scientific Complex "Institute of Applied Systems Analysis" – ESC "IASA", The National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
<https://orcid.org/0000-0003-2327-156X>, wrogoza@wi.zut.edu.pl

Yaroslav Zarichnyi, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
<https://orcid.org/0000-0002-7596-5857>, yarik120700@gmail.com

Abstract. The paper examines the issue of improving the efficiency of service implementation in a service-oriented architecture (SOA) by integrating recommendation systems into the object request broker. The authors analyze the key problems that arise during the operation of two components of SOA: the service discovery pattern on the server side and the service registry usage pattern.

For the service discovery pattern, the identified problems include the complexity of configuring the load balancer, the need for individual traffic balancing due to service heterogeneity, difficulties in balancing the

load on the broker, the dynamic nature of request processing, as well as the complexity of monitoring and analyzing requests for anomaly detection and load forecasting.

For the service registry usage pattern, the authors identified problems with unstructured data about services, lack of metadata about their functional capabilities, interfaces, availability, and quality, as well as limitations in service filtering mechanisms, which complicates the search and selection of services.

As a solution, it is proposed to include a recommendation system in the load balancer and service registry. For the load balancer, it will provide traffic distribution optimization, automatic anomaly detection, load forecasting, resource utilization optimization, and adaptive management of request processing processes.

For the service registry, the recommendation subsystem will allow dynamic updating of recommendations based on demand, grouping users by interests, providing context-sensitive recommendations, enriching user profiles, using collaborative filtering, and evaluating recommendation quality based on user feedback.

The integration of recommendation systems can significantly improve the efficiency, performance, and user experience of SOA through personalized recommendations of relevant services that meet their specific needs. The authors consider this an important step in the development of service-oriented architecture.

Key words: service-oriented architecture; recommendation system; load balancing; service registry; personalization; adaptive management.

Вступ

У світі, насиченому інформацією та вибором, рекомендації, що базуються на контенті, стають важливими для індивідуалізованого досвіду. Рекомендаційні системи виникають як ключове рішення в цьому контексті. Спочатку сприйняті як інструменти, що збирають і подають пропозиції, вони поступово перетворилися на комплексні програмні продукти, які використовують різні методології для надання рекомендацій щодо використання програмних застосунків, які найбільшою мірою відповідають вимогам користувачів і змісту вирішуваних задач. У сервісно-орієнтованій архітектурі (англ. service-oriented architecture, SOA) такі застосунки класифікуються як вебсервіси, призначені для сприяння створенню розподілених застосунків із швидкою, економічно ефективною та простою композицією у гетерогенних середовищах [1]. Ці програмні системи розроблені для надання функцій, доступних через вебтехнології з дотриманням встановлених для них стандартів і врахуванням архітектури систем сервісно-орієнтованих обчислень (англ. service-oriented computing, SOC).

Пошук і відбір документів та методів серед їх великої і постійно зростаючої кількості, які найбільш повно відповідають запитам користувачів, нерідко виявляється складним завданням. У сервісно-орієнтованих обчисленнях використовуються вебсервіси як основа для розробки розподілених застосунків з можливою їх композицією. Вебсервіси стають стандартизованими засобами для взаємодії між програмними застосунками різних платформ і структур. Незважаючи на критично важливу роль вебсервісів у SOA, існують проблеми з їх вибором і виявленням через обмеження в стандартах. До того ж процеси вибору й використання веб-служб ускладнюються внаслідок певних недоліків у структурі реєстру UDDI та в мовах опису вебсервісів, таких як описи WSDL [1].

У цій статті виконано аналіз практичних викликів (названих у статті «проблемами»), що виникають під час експлуатації систем сервісно-орієнтованих обчислень, і переваг, які забезпечуються включенням спеціалізованої рекомендаційної системи у складі брокера об'єктних запитань в моделі SOA. Аналіз виконано з посиланнями на відомі ключові властивості рекомендаційних систем, які доводять доцільність їх використання в контексті брокерів об'єктних запитань для покращення функціональних можливостей сервіс-орієнтованих обчислень.

Можливі варіанти включення рекомендаційних систем до складу брокера об'єктних запитань у SOA

На нашу думку, ефективність реалізації послуг у SOA можна підвищити через накладання на компоненти брокера об'єктних запитань SOA додаткових функцій управління, моніторингу та прогнозування навантажень, які можна розглядати як функції спеціалізованих рекомендаційних систем. До такого висновку можна дійти, аналізуючи певні виклики, які виникають під час функціонування двох компонентів SOA: шаблону виявлення послуг на боці сервера та шаблону використання реєстру служб SOA

Шаблон виявлення послуг на боці сервера

Сервісні хореографії – це універсальний підхід до побудови розподілених систем на основі сервісів. У літературі можна знайти багато підходів, що стосуються різних аспектів хореографії обслуговування, таких як реалізованість хореографії та перевірка відповідності, розподілена координація, формальні хореографічні мови та масштабованість [2]. Як відомо, шаблон виявлення послуг на боці сервера, який є складником стандартної SOA, працює таким чином: клієнт ініціює запит до служби через балансувальника навантаження. Балансувальник навантаження, зі свого боку, запитує реєстр служби та спрямовує кожен запит до доступного екземпляра служби [3]. Так само, як виявляються замовлення на послуги на боці клієнта, екземпляри служби проходять процеси реєстрації та скасування реєстрації в реєстрі послуг.

Використання шаблону пошуку послуг у традиційній SOA викликає певні труднощі та проблеми в реалізації функцій, покладених на системи сервісно-орієнтованих обчислень. Визначимо деякі з них.

Складність налаштування. У середовищі сервісно-орієнтованих обчислень використовуються різні типи сервісів із різними характеристиками та вимогами до навантаження. Налаштування балансувальника навантаження вимагає уважного аналізу цих характеристик і встановлення правильних параметрів для оптимального розподілу трафіку між сервісами. Однак з урахуванням динамічної природи середовища SOA налаштування може стати складним завданням, оскільки потребує постійного моніторингу й адаптації до змін [4].

Необхідність застосування індивідуального підходу до балансування трафіку обробки об'єктних запитань. Ця необхідність обумовлена гетерогенністю сервісів. У SOA можуть працювати різні типи сервісів, такі як мікрослужби, монолітичні застосунки, сервери баз даних тощо. Кожен із цих сервісів може мати власні характеристики, такі як пропускна здатність, час відповіді, завантаженість тощо, які потребують індивідуального підходу до балансування трафіку [5].

Необхідність подолання проблем балансування навантаженням брокера. Підтримка масштабування є важливою функцією балансувальника навантаження в SOA. Зі збільшенням обсягу трафіку або кількості сервісів балансувальник повинен забезпечувати рівномірне розподілення навантаження й уникати перевантаження окремих елементів інфраструктури. Зазначені проблеми є особливо нагальними під час обробки дуже великих обсягів запитів і сервісів, які потенційно можуть відповідати вимогам цих запитів. Вирішення цієї проблеми напряду пов'язане з можливостями брокера передбачати надходження до системи нових запитів. На жаль, в існуючих SOA відсутні програмні засоби автоматичного балансування навантаження.

Труднощі, обумовлені динамічністю процесів обробки запитань. У середовищі SOA сервіси можуть бути додані, змінені або видалені в будь-який момент. Балансувальник навантаження повинен швидко реагувати на запитання, які надходять до системи від користувачів, і адаптуватися до нової конфігурації сервісів, тому автоматизація процесів розгортання та масштабування сервісів вимагає впровадження ефективних програмних засобів планування виконання запитів.

Складність виконання моніторингу й аналізу запитань. Балансувальник навантаження повинен постійно здійснювати моніторинг навантаження на всіх сервісах у SOA. Моніторинг передбачає оцінку швидкості відповіді, обсягу запитів та стану ресурсів і дає змогу вчасно виявляти перевантаження та реагувати на них. Поряд із моніторингом балансувальник навантаження повинен аналізувати трафік, який проходить через нього. Реалізація згаданих функцій має бути спрямована на виконання аналізу типів запитів і динаміки зміни обсягу трафіку, а також виявлення несправностей або аномальної активності окремих сервісів. На основі зібраної статистики балансувальник навантаження може бути в стані робити прогнози щодо майбутнього навантаження. Це дає змогу підготуватися до очікуваного зростання трафіку та вчасно приймати заходи для масштабування інфраструктури. Отже, для ефективного моніторингу й аналізу бажано використовувати автоматизовані засоби. Це можуть бути системи моніторингу й управління, які автоматично збирають дані про стан системи, виявляють аномалії та надсилають сповіщення в разі виникнення проблем. Для більш глибокого аналізу трафіку та виявлення тенденцій доцільно інтегрувати балансувальник навантаження із засобами аналізу даних, такими як, скажімо, системи BI (англ. Business Intelligence) або аналітичні інструменти, спрямовані на подолання зазначених проблем. Це дає змогу отримувати додаткові інсайти та робити точніші прогнози. На основі зібраних даних балансувальник навантаження може оптимізувати використання ресурсів, наприклад розподіляти трафік таким чином, щоб мінімізувати використання обчислювальних ресурсів або максимізувати швидкість відповіді для користувачів [3, 6].

Таким чином, можна стверджувати, що існує необхідність включення рекомендаційної системи до складу балансувальника, до функцій якої входитиме реалізація обчислювальних процесів, скерованих на подолання наслідків прояви згаданих вище проблем.

З огляду на можливості рекомендаційних систем [5] додавання рекомендаційної системи до складу балансувальника навантаження дасть змогу покласти на неї зазначені нижче функції.

Необхідність оптимізації розподілу трафіку. Рекомендаційна система може аналізувати різноманітні параметри, такі як пропускна здатність, завантаженість, час відповіді та стан ресурсів кожного сервісу. На основі цього аналізу вона може виробляти індивідуальні рекомендації для кожного запиту, враховуючи його характеристики та вимоги. Рекомендаційна система може використовувати дані про структуру SOA та взаємозв'язків між різними сервісами для оптимізації розподілу трафіку.

Доцільність автоматичного виявлення аномалій у викликах сервісів. Рекомендаційна система може використовувати різні методи аналізу, такі як статистичні моделі, машинне навчання й аналіз великих даних, для виявлення аномальної активності брокера запитань. Це дає змогу виявляти несподівані зміни в паттернах трафіку, несправності в роботі сервісів або спроби злову системи. Крім того, на рекомендаційну систему доцільно покласти функції відслідковування подій і сповіщення про аномалії поведінки брокера, що дає можливість операторам швидко реагувати на них.

Необхідність прогнозування навантаження. Рекомендаційна система цілком здатна використовувати дані з історії трафіку для розробки моделей прогнозування майбутнього навантаження. Вона може враховувати фактори, які впливають на зміну трафіку, такі як день тижня, час доби, сезонність або маркетингові акції, унаслідок виконання цих дій рекомендаційна система набуває здатності регулярно оновлювати свої прогнози на основі нових даних і змін у середовищі.

Необхідність оптимізації використання ресурсів. Рекомендаційна система може аналізувати доступні ресурси та їхні обмеження, такі як обсяг пам'яті, кількість ядер процесора або пропускна здатність мережі, і рекомендувати оптимальні параметри конфігурації балансувальника навантаження, щоб максимально використовувати доступні ресурси й уникнути перевантаження.

Необхідність адаптивного управління процесами обробки запитань. Рекомендаційна система може використовувати зворотний зв'язок для оцінки ефективності своїх рекомендацій та адаптації до змін у середовищі. Це дає можливість автоматично коригувати стратегії розподілу трафіку та параметри конфігурації на основі отриманих результатів. Функції адаптації управління передбачають застосування процедур навчання, яке відбувається на основі досвіду, і таким чином можна поступово покращувати продуктивність та ефективність управління балансувальником навантаження.

Шаблон використання реєстру служб

Ключову роль у виявленні служб SOA відіграє реєстр служб, функціонуючи як база даних, що містить мережеві розташування примірників служб. Забезпечення високої доступності й ефективності пошуку служб, які відповідають запитам клієнтів, у режимі реального часу має вирішальне значення для ефективного функціонування реєстру послуг. Хоча клієнти можуть кешувати мережеві розташування, отримані з реєстру, ця інформація неминуче застаріває, що призводить до того, що клієнти не завжди можуть виявити потрібні їм екземпляри служб. Отже, реєстр послуг містить кластер серверів, які використовують протокол реплікації для підтримки узгодженості [4]. Використання існуючого шаблону часто викликає зазначені нижче проблеми.

Неструктурованість даних. Ця проблема виникає, коли інформація про послуги в сервісі реєстрацій не є структурованою або не відповідає певним стандартам. Різні сервіси можуть бути зареєстровані в реєстрі з використанням різних форматів даних. Наприклад, один сервіс може мати одну структуру метаданих або формат опису, а другий сервіс – іншу. Унаслідок такої невідповідності ускладнюється процес аналізу й обробки цих даних. Крім того, неструктурованість даних ускладнює їх пошук, аналіз та інтерпретацію. Це може призводити до втрати часу й ресурсів на пошук потрібної інформації та її подальший аналіз, а неструктурованість даних може призводити до помилок і неточностей у їхньому аналізі та використанні. Наприклад, відсутність стандартизованого формату може призвести до неправильного розуміння метаданих та їхнього невірної використання.

Відсутність метаданих. Однією з ключових проблем є відсутність або недостатня повнота метаданих про сервіси. Метадані можуть містити інформацію про функціональні можливості, інтерфейси, доступність, якість та інші характеристики сервісів. Брак цих даних викликає ускладнення у пошуку, виборі та використанні сервісів. Метадані про функціональні можливості сервісів допомагають користувачам розуміти, які конкретні функції або послуги надає кожен сервіс. Метадані містять опис конкретних операцій, які можуть бути виконані, а також опис параметрів, потрібних для виконання цих операцій та значень, повернутих після обробки запитів.

З описом метаданих пов'язана проблема побудови інтерфейсів. Використання відповідним чином побудованих інтерфейсів сприяє більш точній інтерпретації вимог користувачів до сервісів і сервісів між собою. Така інформація може містити формати даних, протоколи комунікації та доступні способи виклику сервісу. Метадані про доступність сервісів вказують на те, коли і за яких умов сервіс може бути доступний для використання. Зокрема, опис метаданих доцільно супроводжувати інформацією про час доступності, обмеження на кількість одночасних викликів або потребу автентифікації, а інформація про характеристики сервісів дає змогу оцінити їхню ефективність, надійність та швидкодію. Такі відомості можуть містити метрики продуктивності, рівень сервісу, обмеження на обробку даних та інші характеристики [6].

Проблема пошуку та вибору сервісів. Ця проблема виникає в сервісно-орієнтованих обчисленнях унаслідок у разі відсутності ефективних механізмів фільтрування сервісів. Наприклад, відсутність можливості фільтрувати сервіси за типом, функціональністю або рівнем доступності може призвести до труднощів у пошуку сервісів, що задовольняють конкретним вимогам користувача. Без можливості сортувати й ранжувати сервіси за певними критеріями користувачам нерідко важко зорієнтуватися серед великої кількості результатів пошуку. Скажімо, користувачам може бути потрібно сортувати сервіси за рівнем якості, швидкістю або надійністю, щоб знайти найкращий сервіс для своїх потреб. Якщо інформація про сервіси є неповною або взагалі відсутня, то пошук відповідних сервісів стає значною мірою непередбачуваним навіть у разі, коли користувач точно визначає вимоги до послуг, які йому потрібні. Наприклад, якщо в метаданих відсутні повні відомості про функціональність або параметри сервісів, користувачам буде важко оцінити, чи вони відповідають їхнім потребам [7, 8].

Крім того, якщо дані про сервіси в реєстрі не представлені у формі однорідної структури або формату, це ускладнює їх оброблення та порівняння користувачами навіть у випадках, коли опис вимог до сервісів чітко визначений користувачем і дані про параметри сервісів чітко визначені у відповідних описах сервісів.

Подолання окреслених вище труднощів може бути покладено на спеціалізовану певним чином рекомендаційну систему з додатковими функціями і не вимагатиме реконструкції вже існуючих SOA. На підставі наведених міркувань можна запропонувати модель SOA, у якій рекомендаційна система входить як підсистема сервісу реєстрацій послуг (рис. 1).

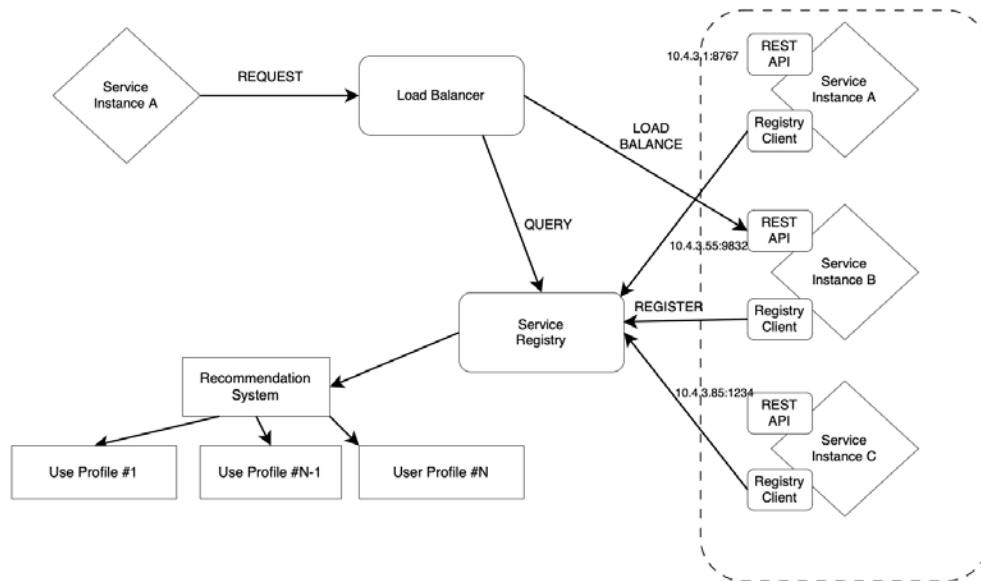


Рис. 1. Рекомендаційна система в складі сервісу реєстрацій послуг

Таким чином, поліпшення роботи сервісу реєстрацій завдяки застосуванню рекомендаційної підсистеми та створенню профілів користувачів може стати персоналізованим і сприяти підвищенню ефективності процесу пошуку та вибору сервісів з більш точним врахуванням потреб користувачів.

Можна виділити такі переваги запропонованої моделі, які впливають завдяки використанню рекомендаційної підсистеми.

Динамічне оновлення рекомендацій. Рекомендаційна підсистема може постійно аналізувати зміни в попиті на сервіси та динамічно оновлювати рекомендації відповідно до цих змін. Наприклад, якщо певний сервіс стає популярним серед користувачів, він може бути рекомендований іншим користувачам, які мають подібні вимоги.

Групування користувачів за інтересами. Використання рекомендаційної підсистеми може дати змогу створювати групи користувачів за їхніми інтересами та потребами. Наприклад, користувачі, які виявляють інтерес до певних типів сервісів, можуть бути об'єднані в спільні групи для зручності надання персоналізованих рекомендацій.

Контекстно-чутливі рекомендації. Рекомендаційна підсистема може враховувати контекстні фактори, такі як геолокація, час доби або пристрій, на якому виконується запит, для надання більш точних і релевантних рекомендацій. Наприклад, користувачам, які використовують мобільні пристрої, можуть бути рекомендовані послуги, оптимізовані саме з врахуванням згаданої мобільності пристроїв [9].

Збагачення профілів користувачів. Сервіс реєстрацій може збагачувати профілі користувачів за допомогою додаткових даних, таких як соціальні мережі, історія переглядів або попередні запити. Це дасть змогу створювати більш деталізовані та повні профілі, що поліпшить якість рекомендацій.

Колаборативна фільтрація. Використання колаборативної фільтрації дає змогу враховувати попередні вибори та поведінку користувачів, а також подібність їхніх виборів до інших користувачів. Цей підхід може сприяти формулюванню більш точних персоналізованих рекомендацій [10].

Оцінка якості рекомендацій. Сервіс реєстрацій може мати механізми для оцінки ефективності рекомендацій з огляду на реакції користувачів на них. Наприклад, користувачі можуть залишати відгуки або оцінки після використання рекомендованих сервісів, що допомагає вдосконалювати якість рекомендацій в майбутньому.

Висновки

Використання рекомендаційної підсистеми в архітектурі SOC може значно поліпшити ефективність, продуктивність і користувацький досвід системи. Рекомендаційна підсистема допомагає користувачам швидко знаходити необхідні сервіси за допомогою персоналізованих рекомендацій, що відповідають їхнім уподобанням і потребам. Користувачі можуть отримувати рекомендації щодо найбільш підходящих сервісів для їхніх конкретних завдань або сценаріїв використання. За допомогою рекомендаційної підсистеми можуть створюватися персоналізовані профілі користувачів і надаватися рекомендації, які враховують їхні індивідуальні вподобання, історію взаємодій та контекст.

Шляхом рекомендацій користувачі можуть знаходити нові для себе сервіси або використовувати додаткові функціональності, які можуть бути корисними для їхніх поточних потреб. Рекомендаційні

системи можуть також допомогти користувачам вибирати найновіші й найактуальніші версії сервісів, що дає змогу підтримувати систему в сучасному стані. Таким чином, завдяки використанню рекомендаційної підсистеми у SOA можна підвищити зручність, ефективність і користувацькі вимоги від використання сервісів, що є певним кроком у розвитку цієї архітектури.

Література

1. Fabrizio Ciacetta, Giovanni Bucci, Carmine Landi. Taxonomies of distributed measurement systems via UDDI registry. IEEE Instrumentation and Measurement Technology Conference, 2008, pp. 1316–1321.
2. Filippone G., Pompilio C., Autili M. & Tivoli M. An architectural style for scalable choreography-based microservice-oriented distributed systems. Computing, 105, 2023, pp. 1933–1956.
3. Chris Richardson, Service Discovery in a Microservices Architecture, 2015. URL: <https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/> (дата звернення 20.02.2024).
4. Simone Cusimano, Service Discovery in Microservices, 2022. URL: <https://www.baeldung.com/cs/service-discovery-microservices> (дата звернення 15.02.2024).
5. Karolina Holewa, Perks of recommendation systems in business, 2023. URL: <https://www.miquido.com/blog/perks-of-recommendation-systems-in-business/> (дата звернення 20.02.2024).
6. An SOA Design Patterns Recommendation System Based on Ontology URL: https://link.springer.com/chapter/10.1007/978-3-030-16657-1_95 (дата звернення 01.03.2024).
7. Ezdehar Jawabreh, Adel Taweel. Time-Aware QoS Web Service Selection Using Collaborative Filtering. A Literature Review, pp. 55–69.
8. Dang Y. Personalized Recommendation of Literature Resources in University Library Based on Abstract Content Filtering Algorithm, 2023.
9. Felfernig A., Atas M., Helic D., Tran T. N. T., Stettinger M., & Samer R. Algorithms for Group Recommendation. In Group Recommender Systems, 2024, pp. 29–61.
10. Trattner C., Said A., Boratto L., Felfernig A. Evaluating Group Recommender Systems. In Group Recommender Systems, 2024, pp. 63–75.

Reference

1. Fabrizio Ciacetta, Giovanni Bucci, Carmine Landi (2008). Taxonomies of distributed measurement systems via UDDI registry, IEEE Instrumentation and Measurement Technology Conference, pp. 1316–1321.
2. Filippone, G., Pompilio, C., Autili, M., & Tivoli, M. (2023). An architectural style for scalable choreography-based microservice-oriented distributed systems. Computing, 105, pp. 1933–1956.
3. Chris Richardson (2015). Service Discovery in a Microservices Architecture [Online]. Available: <https://www.nginx.com/blog/service-discovery-in-a-microservices-architecture/>.
4. Simone Cusimano (2022). Service Discovery in Microservices [Online]. Available: <https://www.baeldung.com/cs/service-discovery-microservices>.
5. Karolina Holewa (2023). Perks of recommendation systems in business [Online]. Available: <https://www.miquido.com/blog/perks-of-recommendation-systems-in-business/>.
6. An SOA Design Patterns Recommendation System Based on Ontology [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-16657-1_95.
7. Ezdehar Jawabreh, Adel Taweel. Time-Aware QoS Web Service Selection Using Collaborative Filtering. A Literature Review, pp. 55–69.
8. Dang, Y. (2023). Personalized Recommendation of Literature Resources in University Library Based on Abstract Content Filtering Algorithm.
9. Felfernig, A., Atas, M., Helic, D., Tran, T. N. T., Stettinger, M., & Samer, R. (2024). Algorithms for Group Recommendation. In Group Recommender Systems, pp. 29–61.
10. Trattner, C., Said, A., Boratto, L., Felfernig, A. (2024). Evaluating Group Recommender Systems. In Group Recommender Systems, pp. 63–75.