

УДК 303.064

DOI <https://doi.org/10.32782/tnv-tech.2024.2.4>

ОГЛЯД ХОРЕОГРАФІЇ ВЕБ-СЕРВІСІВ WSMO ДЛЯ ВИКОНАННЯ СИНХРОННИХ ТА АСИНХРОННИХ ЗАПИТІВ

Касьянчук І. В. – аспірант

Національного технічного університету України

«Київський політехнічний інститут імені Ігоря Сікорського»

ORCID ID: 0009-0000-2215-149X

Популярність Інтернету зумовлює стрімке зростання обсягів його інформаційного наповнення. Дані, які надають веб-ресурси орієнтовані здебільшого на розуміння людиною, і лише невелика їх частина придатна для обробки програмними агентами. Через це виникає складність у пошуку потрібної інформації, послуг або сервісів тощо.

Це зумовило потребу у стандартизації ресурсів, а саме – уніфікації обміну даними між різними сервісами, чи інтегрування об'єктів реального світу в мережу Інтернет. Концепція семантичного Web (Semantic Web) дає можливість пов'язувати сервіси за допомогою опису взаємодії, включаючи пошук потрібних сервісів за запитом (Service Discovery). Таким чином досягається «розуміння» інформації не лише людиною, а й системами для подальшої автоматичної обробки програмами інших виробників, після чого вона передається користувачу.

Другою концепцією, яка доповнює Семантичний Web є теоретичне уявлення про моделі предметних областей, або онтологію. Стандартом мови опису консорціумом W3C визначено XML (eXtensible Markup Language), що являє собою документ з переліком тегів та атрибутів, визначених певним протоколом або специфікацією. Для онтологій однією з таких специфікацій визначено RDF (Resource Definition Framework) – абстрактна мова метаданих, сформованих у вигляді графу. Консорціумом рекомендовано багато фреймворків для здійснення безпосереднього опису семантичної моделі, одним з прикладів є OWL (Web Ontology Language), що визначає мережу понять та відносини між ними. Іншою важливою адаптацією до програмування сервісних додатків є WSMO. Цей фреймворк підтримує безпосереднє виконання Service Discovery – викриття потрібних сервісів для виклику за запитом користувача, а також хореографію разом з оркестрацією, що робить його більш досконалим та повним рішенням у порівнянні з OWL-S.

У статті розглядаються підходи, запропоновані фреймворком WSMO для оркестровки та хореографії веб-сервісів, а також недоліки цього рішення у контексті обробки нескінченних потоків даних (data streams).

Ключові слова: сервіс-орієнтована архітектура, онтологія, WSMO, семантичний опис веб-сервісів.

Kasianchuk I. V. Review of the web service choreography concept in WSMO for executing synchronous and asynchronous requests

The popularity of the Internet has led to a rapid increase in its informational content. However, much of the data provided by web resources is primarily oriented towards human understanding, with only a small portion suitable for processing by software agents. This complexity poses challenges in finding the necessary information, services, or resources.

To address this, there is a need for standardization of resources, specifically the unification of data exchange between different services, or the integration of real-world objects into the Internet. The concept of the Semantic Web enables the linking of services through interaction descriptions, including service discovery. This facilitates the "understanding" of information not only by humans but also by systems for further automated processing by other programs, before being transmitted to the user.

The Semantic Web is complemented by the theoretical concept of domain models, or ontologies. The W3C consortium has defined XML (eXtensible Markup Language) as the standard language for describing documents with a list of tags and attributes defined by a specific protocol or specification. For ontologies, one of the specifications defined is RDF (Resource Description Framework) – an abstract language of metadata formed as a graph. The consortium recommends many frameworks for directly describing the semantic model, one example being OWL (Web Ontology Language), which defines a network of concepts and relations between them.

Another important adaptation for programming service applications is WSMO (Web Service Modeling Ontology). This framework supports direct execution of service discovery – the exposure of necessary services for invocation by user request, as well as choreography and orchestration, making it a more comprehensive solution compared to OWL-S.

The article discusses approaches proposed by the WSMO framework for the orchestration and choreography of web services, as well as the drawbacks of this solution in the context of processing infinite data streams.

Key words: *service-oriented architecture, ontology, WSMO, semantic description of web services.*

Вступ. Процеси створення програмних продуктів вимагають залучення спеціалістів в галузі інформаційних технологій. Стрімкий розвиток системного проектування, що підтримує найрізноманітніші протоколи та мови для опису вимог, дозволяє звести цю залежність від постачальників програмного забезпечення до мінімуму.

Важливим кроком для цього є перенесення об'єктів реального світу у всевітню мережу Інтернет, та відповідна стандартизація для взаємодії з ними за допомогою веб-сервісів. Метою є створення сервісів загального призначення, які, за описом моделі запиту та відповіді, надають необхідну для користувача інформацію.

Дослідження останніх тенденцій серед фреймворків для опису предметної області сервісів (онтологій) вказують на декілька великих концептуальних рішень, серед яких варто виділити OWL та WSMO. Обидва фреймворки працюють за «об'єктно-орієнтованою» моделлю програмування, тобто концепції, при якій програма розглядається як множина об'єктів, які взаємодіють між собою. Підхід побудований на використанні типів даних, класів, атрибутів та відношень між ними.

Постановка проблеми. Незважаючи на переваги, здебільшого пов'язані з схожістю подання сутностей об'єктно-орієнтованого програмування з реальним світом, цей підхід не виключає складностей та помилок при реалізації. Один з недоліків впливає з обсягів сучасного користування мережею Інтернет: при надмірному навантаженні на сервіс, неправильне архітектурне планування може призвести до його виходу з ладу. Причиною цього є використання багатопотоковості, через нехтування якою виникають проблеми: стан гонитви (*race conditions*), взаємні блокування (*deadlocks*) тощо.

Основна частина. Семантичні веб-сервіси дозволяють оперувати логікою за допомогою розуміння сенсу запиту. Використання синтаксичних каркасів, таких як XML, дозволило представляти та публікувати машинночитаемі, декларативні специфікації, які можуть бути використані як розробниками, так і додатками. Цей підхід на основі стандартів загалом вважається одним із найкращих способів реалізації повного потенціалу Вебу та його пов'язаних технологій. Шляхом розробки відкритих протоколів та рекомендацій програмні компоненти можуть легко взаємодіяти один з одним на різних платформах та різних мовах стандартизованим способом.

WSMO є одним з найбільш значущих фреймворків семантичних веб-сервісів, запропонованих до цього часу. Він доповнює існуючі синтаксичні стандарти веб-сервісів, надаючи концептуальну модель і мову для семантичного маркування всіх відповідних аспектів загальних веб-сервісів. Характеристики сервісу визначаються через опис, статичної семантики та динамічної семантики у кількох документах з урахуванням метамоделі WSMO.

WSMO виступає в якості метамоделі для таких сервісів на основі мета-об'єктного механізму (*meta-object facility*). Семантичні описи сервісів, згідно

з метамоделлю WSMO (рис. 1), можуть бути визначені за допомогою однієї з кількох формальних мов, визначених як WSML (*Web Service Modelling Language*), і складаються з чотирьох основних елементів, які вважаються необхідними для підтримки семантичних веб-сервісів, а саме: онтології (*ontology*), цілі (*goal*), веб-сервіси (*web service*) та посередники (*mediator*).

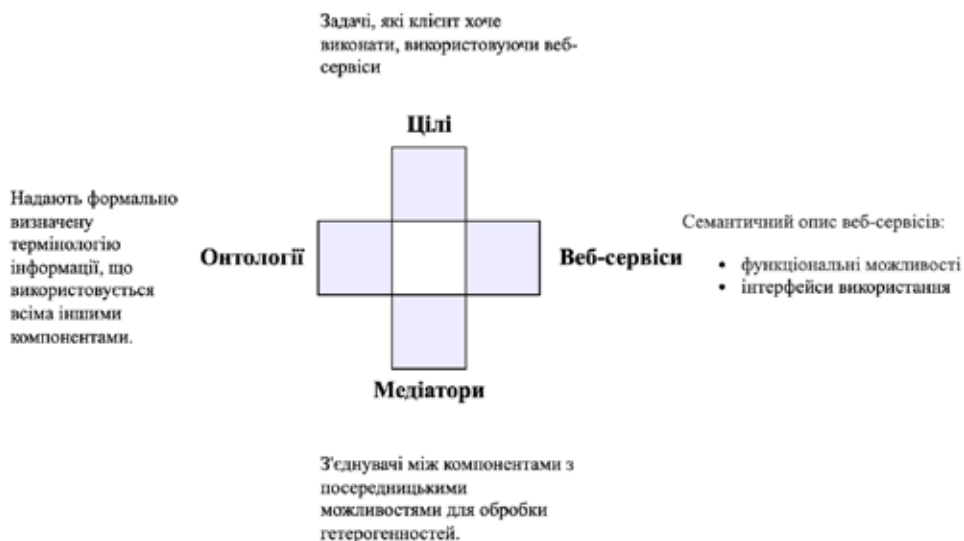


Рис. 1. Структура WSMO

Онтології описуються у WSMO на мета-рівні. Мета-онтологія підтримує опис всіх аспектів онтологій, які надають термінологію для інших елементів WSMO. Типова онтологія складається з нефункціональних властивостей (з існуючого набору попередньо визначених основних властивостей), імпортованих онтологій та визначення понять, відносин, аксіом, функцій та екземплярів онтології. Крім того, вона вказує оо-посередники (*ooMediators*), які використовуються для імпорту онтологій, які вирішують проблеми підлаштування, злиття або перетворення [1].

У WSMO визначаються **цїлі**, які можуть бути у клієнта при взаємодії з веб-сервісом. Вони складаються з нефункціональних властивостей, імпортованих онтологій, використаних посередників, післяумов та ефектів. Післяумови та ефекти описують стан інформаційного середовища, що потребується замовником. Онтології можуть бути безпосередньо імпортовані як термінологія для визначення цілі, коли не потрібно вирішувати конфлікти (наприклад, неспівпадіння схеми відповіді тощо). Однак, якщо необхідне вирівнювання, злиття або вирішення конфліктів, вони імпортуються через посередників типу *ooMediators*.

Описи **веб-сервісів** у WSMO надають семантичний опис самостійних веб-сервісів з анотаціями, включаючи їх функціональні та нефункціональні властивості, а також інші аспекти, необхідні для взаємодії з ними. Необхідна термінологія, як і для цілей, може бути імпортована безпосередньо або через посередників типу *ooMediators*, коли потрібно вирішити конфлікти. Крім того, описується можливість (*capabilities*) та інтерфейси (*interfaces*) сервісу: можливість визначає

функціональні аспекти запропонованого сервісу, моделювані в термінах передумов (*preconditions*), припущень (*assumptions*), післяумов (*postconditions*) та ефектів (*effects*); тоді як інтерфейси представляють інтерфейси, орієнтовані на дані, сервісу (наприклад, ті, що визначені за допомогою WSDL), та надають деталі щодо його функціонування з точки зору його *хореографії* та *оркестрації* [1].

Поняття **Посередників** було введено WSMO з метою вирішення проблем гетерогенності, і вони є спеціальними елементами, що використовуються для зв'язку гетерогенних компонентів, що беруть участь у моделюванні веб-сервісу. Вони визначають необхідні відображення, трансформації або зведення між зв'язаними елементами. Чотири різновиди посередників визначені як *ggMediators*, *ooMediators*, *wgMediators* і *wwMediators*:

- *ggMediators*: ці посередники з'єднують дві *цілі*, виражаючи зведення джерела цілі до цілі призначення. Вони можуть використовувати *ooMediators* для обходу відмінностей в термінології, що використовується для визначення цих цілей. Крім того, WSMO дозволяє зв'язувати не лише цілі, але й цілі з *ggMediators*, що дозволяє перевикористовувати кілька цілей для визначення нової;
- *ooMediators*: ці посередники імпортують онтології та вирішують можливі неузгодженості у представленні між ними, такі як різниці у мовах представлення або у концептуалізації того ж домену;
- *wgMediators*: вони зв'язують веб-сервіс з ціллю. Цей зв'язок представляє (повне або часткове) виконання цілі веб-сервісом. *wgMediators* можуть використовувати *ooMediators* для вирішення проблем гетерогенності між веб-сервісом та ціллю;
- *wwMediators*: ці посередники зв'язують два веб-сервіси, містять *ooMediators*, необхідні для подолання проблем гетерогенності, які можуть виникати в разі, коли сервіси використовують різні словники.

WSMO використовує F-Logic як основний семантичний фреймворк для опису концепцій та описів сервісів [1].

Опис веб-сервісів у WSMO визначається через можливості (*capabilities*) та інтерфейси (*interfaces*), які в свою чергу діляться на оркестровку (*orchestration*) та хореографію (*choreography*).

Веб-сервіс повинен мати точно одну **можливість**, яка визначає функціональність сервісу. Можливість веб-сервісу визначається за допомогою вказівки *передумови*, *післяумови*, *припущення* та *ефекту*, кожне з яких є набором виразів. Можливість веб-сервісу також оголошує набір змінних, які спільно використовуються між виразами. Терміни, використані в цих виразах, повинні бути формально визначені в деяких онтологіях, які мають бути імпортовані або безпосередньо, або через посередників типу *ooMediators*. Можливість, а отже, і веб-сервіс можуть бути пов'язані з певними цілями, які вирішуються веб-сервісом за допомогою спеціальних типів посередників, що називаються *wgMediators*.

Інтерфейс описує, як функціональність веб-сервісу може бути досягнута з хореографічного та оркестрального погляду. Вони визначають, як використовується інтерфейс при приєднанні до кількох інших сервісів.

Оркестрація – це декларативна специфікація, яка описує робочий процес для підтримки виконання конкретного бізнес-процесу, операції або сервісу. Мови, такі як BPEL4WS, описують як потік даних, так і потік процесу через робочий процес; тобто, патерн взаємодій, який агент веб-сервісу повинен виконати, а також поняття, такі як транзакції та відкат (у випадку невдачі сервісу).

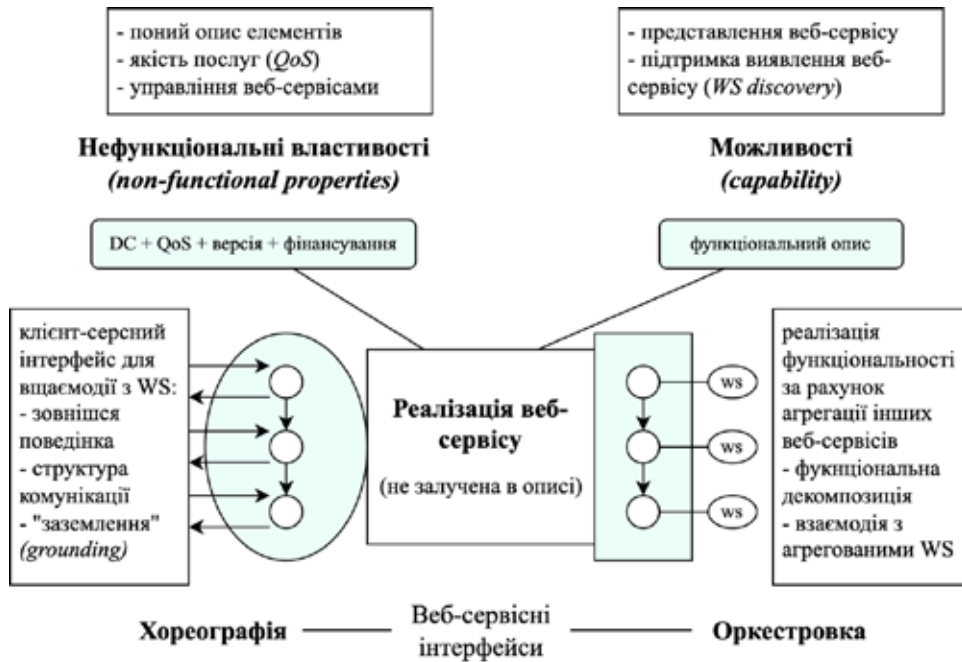


Рис. 2. WSMO опис веб-сервісу

Наприклад, оркестрація процесу ланцюга постачання продуктів може описувати бізнес-протокол, який формалізує: 1) інформаційні елементи, які може містити замовлення продукту; 2) порядок (та посилання на інтерфейси, визначені у WSDL), яким ці елементи повинні бути надані; та 3) які винятки можуть бути оброблені. Оркестрації зазвичай виконуються одним рушієм робочого процесу, який відповідає за керування комунікацією та потоком даних між сервісами. Роль оркестрації була значним чинником у розвитку як електронних бізнес-процесів, так і робочих процесів у сфері електронної науки, і багато напівавтоматизованих інструментів та робочих станцій виникли для сприяння побудові цих робочих процесів.

Хореографія визначає послідовність та умови, за яких кілька співпрацюючих, незалежних агентів обмінюються повідомленнями для виконання завдання з досягненням цільового стану. Мови, такі як WS-Choreography (яка базується на Pi-Calculus), описують взаємодії зі сповіщаючою стороною (якою можуть бути інші веб-сервіси, додатки або люди) та можуть складатися з кількох окремих взаємодій, чие поєднання становить повну транзакцію. Це поєднання, разом з її протоколами повідомлень, інтерфейсами, послідовностями та пов'язаною логікою, розглядається як хореографія (рис. 2).

Таким чином, для узагальнення, хореографія описує зразок комунікації, який дозволяє користуватися функціональністю веб-сервісу іншими клієнтами, у той час як оркестровка описує, як різні веб-сервіси повинні взаємодіяти для досягнення загальної функціональності веб-сервісу. Отже, інтерфейс хореографії в цьому прикладі має три правила переходу з урахуванням визначень термінології.

Наступний приклад описує хореографію системи мовою WSMO для бронювання квитків на літак. З метою спрощення, ми припускаємо наступну поведінку:

спочатку сервіс очікує запит на бронювання квитка на поїздку поїздом, яка починається в Австрії. Якщо це так, він створює тимчасову бронь для цього квитка. Наступним кроком є очікування інформації про кредитну картку для оплати: якщо кредитна картка, надана клієнтом, є дійсною, створюється бронювання, інакше бронювання скасовується [2].

```
choreography BookTicketServiceChoreographyInterface
importsOntology{ _"http://example.org/tripReservationOntology",
  _"http://example.org/purchaseOntology"}

vocabulary_in hasValue
  {tr#reservationRequest, tr#tripFromAustria, tr#ticket}
vocabulary_out hasValue
  {tr#reservation, tr#negativeAcknowledgement}
vocabulary_shared hasValue
  {tr#temporaryReservation, po#creditCard}

guardedTransitions BookTicketChoreographyTransitionRules

if (reservationRequestInstance [
  reservationItem hasValue ?trip,
  reservationHolder hasValue ?reservationHolder
] memberOf tr#reservationRequest and
?trip memberOf tr#tripFromAustria and ticketInstance [
  trip hasValue ?trip,
  recordLocatorNumber hasValue ?rln
] memberOf tr#ticket
then
temporaryReservationInstance[
  reservationItem hasValue ticketInstance,
  reservationHolder hasValue ?reservationHolder
] memberOf tr#temporaryReservation
...
```

Як видно з опису концепцій взаємодії, сервіси повинні мають бути централізовано зареєстровані, тобто мати певну базу знань, яка містить описи всіх сервісів: онтологію запитів, перед- та післяумов, ефектів тощо. Хореографія вимагає чітко визначеного запиту від клієнта для подальшого виконання бізнес-логіки. Сучасні додатки, які оброблюють потоки даних великого обсягу працюють за асинхронною моделлю, наприклад використовуючи чергу повідомлень (рис. 3). Сервіси, залучені для хореографії виступають споживачами (*subscribers*) даних, які очікують на нове повідомлення з черги від постачальника (*publishers*). Подальша оркестровка дозволяє обробити ці дані, та помістити відповідь до іншої відповідної черги, зберегти її до своєї бази даних тощо.

Крайня версія специфікації WSMO-Lite опублікованої у 2010 році містить елементи для опису хореографії притаманні для синхронної взаємодії веб-сервісів. Гнучкість специфікації обмежується посередниками типу *wwMediators*, які адаптують виклики між двома сервісами, але не здатні аналізувати фрейми даних або дані з брокера повідомлень у вигляді черги.

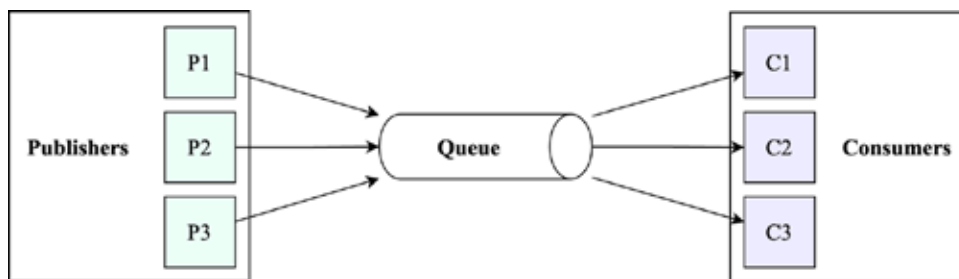


Рис. 3. Хореографія веб-сервісів за допомогою черги

Оскільки WSMO є одним з повноцінних доробок у сфері семантичних описів веб-сервісів, можна дійти висновку що інші фреймворки так само не мають здатності адаптації свого опису до реактивних систем або потоків даних.

Висновки. У роботі було розглянуто можливості семантичних фреймворку для опису онтології веб-сервісів. Використовуючи онтології як модель даних і на основі вичерпних семантичних описових можливостей, фреймворки автоматично виконують виявлення, композицію, конвертацію та виконання веб-сервісів.

Було розглянуто принципи дизайну та мотивацію для Онтологій, Веб-сервісів, Цілей та Посередників як чотирьох основних елементів Семантичних веб-сервісів, а також їх визначення та моделювання в рамках WSMO.

Порушено питання подальшого вдосконалення фреймворків для асинхронної обробки запитів. У сучасному WEB широко використовується обробка великих потоків даних, що зумовлює доповнення специфікації WSMO та інших доробок даної сфери.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:

1. Wang H. A Formal Model of the Semantic Web Service Ontology (WSMO) [Електронний ресурс] / H. Wang, N. Gibbins // 13th IEEE International Conference on Engineering of Complex Computer Systems. Режим доступу до ресурсу: <https://core.ac.uk/download/pdf/287750875.pdf>.
2. Semantic Web Services – Concepts and Technology [Електронний ресурс] / M. Stollberg, C. Feier, D. Roman, D. Fensel // Digital Enterprise Research Institute. Режим доступу до ресурсу: <https://michael-stollberg.de/publications/wsmo-bookchapter.pdf>.
3. WSMO-Lite: Lightweight Semantic Descriptions for Services on the Web [Електронний ресурс] // W3C Member Submission. 2010. Режим доступу до ресурсу: <https://www.w3.org/submissions/2010/SUBM-WSMO-Lite-20100823/>.

REFERENCES:

1. Wang, H., & Gibbins, N. (n.d.). A formal model of Semantic Web Service Ontology (WSMO). Retrieved from <https://core.ac.uk/download/pdf/287750875.pdf>
2. Stollberg, M., Feier, C., & Roman, D. (n.d.). Semantic Web Services – Concepts and Technology. Retrieved from <https://michael-stollberg.de/publications/wsmo-bookchapter.pdf>
3. WSMO-lite: Lightweight semantic descriptions for services on the web. (2010). Retrieved from <https://www.w3.org/submissions/2010/SUBM-WSMO-Lite-20100823/>