

ПРОБЛЕМИ ПОБУДОВИ СЕРВІС-ОРІЄНТОВАНИХ ПРИКЛАДНИХ ІНФОРМАЦІЙНИХ СИСТЕМ В SEMANTIC WEB СЕРЕДОВИЩІ НА ОСНОВІ АГЕНТНОГО ПІДХОДУ

П. Андон, В. Дерезький

Інститут програмних систем НАН України
03187, Київ-187, проспект Академіка Глушкова, 40,
тел.: +38 044 526 4342; {andon, dva}@isofts.kiev.ua

У роботі представлені основні засоби, що покладені в основу технології сервіс-орієнтованої інтеграції. Семантичні технології забезпечують існування певного рівня абстракції над існуючими інформаційними технологіями. Цей рівень дозволяє здійснювати зв'язок даних, семантики і процесів, що виконуються в розподіленому середовищі. Розвиток цих технологій дасть змогу організаціям створювати єдине уніфіковане представлення даних у всіх застосуваннях, дозволить точно знаходити необхідну інформацію, спрощувати корпоративну інтеграцію і інтеграцію мережних застосувань. Запропонована схема для представлення та рішення задач в розподіленому середовищі Семантичної мережі. Розглянуто засоби та стандарти інтеграції та координації мережних сервісів на основі потоку робіт (workflow).

The emerging paradigm of Semantic Web promises to bring to distributed computation and services the flexibility that the web has brought to the sharing of Information Resources. An understanding of fundamental properties of e-service composition is required in order to take full advantage of the paradigm. This paper examines proposals and standards for e-services from the perspectives of XML, data management, workflow, and process models. Key areas for study are identified, including behavioural service signatures, verification and synthesis techniques for composite services, analysis of service data manipulation commands. The offered chart for presentation and decision of tasks in the distributed environment of the Semantic Web. Facilities and standards of integration and co-ordination of Semantic Web services are considered on the basis of stream of works (workflow).

1. Засоби рішення задач в Семантичній мережі

Ініціатива Семантичної мережі (Semantic Web) була сформульована одним із законодавців в області сучасних інформаційних технологій – консорціумом W3C і відразу привернула до себе увагу. Концепцію Семантичної мережі висунув Тім Бернерс-Лі – один з основоположників World Wide Web і голова консорціуму (W3C) в роботі „Semantic Web” [1].

Загальна ідея проекту Семантичної мережі полягає в організації такого представлення даних в мережі, щоб допускалася не тільки візуалізація, але й ефективна автоматична обробка даних програмами різних виробників. Шляхом радикальних перетворень традиційного Web передбачається створення системи семантичного рівня.

За задумом творців Семантичної мережі повинно забезпечуватися „розуміння” інформації комп'ютерами, формування найбільш відповідної до потреб користувача інформації.

Запропонована ідея використовувати знання разом з даними – дійсно революційна. Підходи, які пропонувалися раніше, в яких використовувалися схеми даних і метадані, вирішували лише частково ці задачі і не породжували якісного просування, що намітилося в Семантичній мережі.

Суть ідеї можна представити таким чином: якщо розглянути пару „дані-програми”, що використовується в традиційних програмних системах, то очевидно, що значення оброблюваних даних міститься в самих програмах, які ці дані обробляють, в їх кодї міститься алгоритм розуміння цих даних. У підходах Семантичної мережі вводяться метадані, тобто дані про дані, які розглядаються як знання про дані і які винесені за межі програмного коду. В метадані вкладено смисл даних або частину смислу. Якщо метадані обробляти таким же чином і разом з даними, то програми зможуть стати універсальними та спрощеними, що є прогресивним чинником.

На рис. 1 показана схема, на якій зображені програми, кожна з яких працює з певним набором даних. Програма, окрім спеціальних даних, обробляє дані, що є віддзеркаленням традиційних сутностей реального світу, таких як матеріали, адреси, організації, рахунки і т.д. Назвемо ці дані загальними, на відміну від спеціальних даних, що відображають приватні особливості програми.

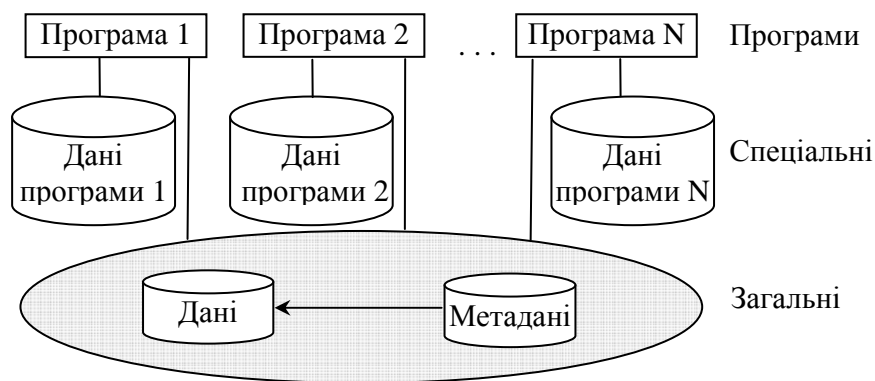


Рис. 1

Природною побудовою інформаційних систем відповідно до даної моделі є умовно дворівневе розбиття даних на загальні і спеціальні. Причому загальні дані можуть мати певну структуру.

Інформаційні системи нового покоління представляються як розподілені системи, що використовують бази даних, які містять загальні „осмислені дані” – моделі світу, предметної області і спеціальні дані, специфічні для конкретної інформаційної системи або специфікації задачі. Сумісність загальних даних та їх описів повинна забезпечуватися загальною методологією і стандартами. Програмний код таких систем повинен бути універсальним, інтерпретуючим довільні специфікації даних або спеціалізованим, сформованим так, щоб враховувати універсальне рішення і потрібний контекст.

Парадигмою для побудови таких програмних систем є підхід, заснований на використанні семантичних Web-сервісів [2]. Семантичні Web-сервіси (SWS) розширюють поняття звичайних Web-сервісів у частині використання семантичної інформації, а саме онтологій та семантичної розмітки як для прикладних, так і для системних потреб [3]. Онтології використовують для кодування класів і підкласів понять і відносин, що мають відношення до сервісів і обмежень користувача. Онтології надають індивідуальному сервісу можливість успадковувати розділені поняття і словники із специфічної області. Наприклад, ми можемо визначити онтологію, що містить клас „Купувати” з підкласом „Купувати мобільний телефон”, що в свою чергу має підкласи „Купувати мобільний телефон Motorola”, „Купувати мобільний телефон Nokia” і т. д. В даний час створюються програми, здатні шукати потрібні сервіси. Такі засоби як UDDI-сервер містять перелік доступних Web-сервісів. І хоча програма або агент може знайти Web-сервіс без допомоги людини, вона не в змозі зрозуміти, як саме ним користуватися і навіть для чого він призначений. Мова опису Web-сервісів (WSDL) дає нам інструмент для опису того, яким чином можна взаємодіяти з тим або іншим Web-сервісом, тоді як семантична розмітка, що є засобом SWS, забезпечує нас інформацією про те, що і як робить даний сервіс [4].

Щоб SW-сервіси стали реальністю, мова розмітки повинна бути досить інформативною у тому сенсі, щоб комп'ютер був здатний самостійно розуміти значення записаних виразів. Вимоги, яким повинна відповідати така мова, полягають в наступному:

- необхідність пошуку сервісів (виявлення – discovery). Програми повинні мати можливість самостійно знаходити необхідні їм Web-сервіси. Слід зауважити, що ні WSDL, ні UDDI не дають можливості програмі зрозуміти, для чого саме з погляду клієнта служить той або інший Web-сервіс. Семантичний Web-сервіс зможе пред'явити опис своїх властивостей і можливостей з тим, щоб програми могли самі розпізнавати його призначення;

- необхідність запускати сервіси (запуск – invocation). Програми повинні вміти самостійно розпізнавати, яким чином потрібно запускати і виконувати даний сервіс. Наприклад, якщо виконання сервісу є багатокроковою процедурою, то програма повинна знати, як їй слід взаємодіяти з сервісом, щоб виконати послідовність кроків. SW-сервіс повинен забезпечити вичерпний перелік того, що повинен уміти агент для запуску і виконання даного сервісу. Він повинен також містити опис вхідних і вихідних даних;

- необхідність спільного використання кількох сервісів (композиція – composition). Програми повинні вміти відбирати потрібні їм Web-сервіси і комбінувати їх для досягнення своєї мети (вирішення задачі). Сервісам необхідно буде тісно взаємодіяти один з одним, таким чином, щоб в результаті їх спільного виконання рішення поставленої задачі було прийнятним. Таким чином, програмні агенти зможуть будувати абсолютно нові сервіси, комбінуючи вже існуючі сервіси в мережі;

- необхідність контролювати стан виконання, що відбувається після запуску сервісу (моніторинг – monitoring). Програмний агент повинен уміти визначати властивості даного сервісу і стежити за його виконанням. Деяким сервісам потрібен певний час для виконання роботи і агенти повинні стежити за ходом виконання сервісу.

Підхід, що пропонується в цій роботі, спрямований на створення нових застосувань шляхом забезпечення агентів можливостями самостійно знаходити, запускати, комбінувати і стежити за виконанням сервісів без участі людини, використовуючи засоби онтологій та семантичні розмітки.

У різних напрямках інформаційних технологій (ІТ) створена велика кількість різних архітектур застосувань з використанням Web-сервісів, що функціонують у різних операційних оточеннях. Використання засобів SWS в цих застосуваннях може надати їм нові якості та переваги, а саме [5]:

- **B2B** – “Бізнес – Бізнес” архітектури застосовуються в системах бізнес інтеграції. Ця область представляє частину розвинутої технології для обслуговування широкого кола запитів, спрямованих на рішення проблем електронної комерції. Це один з головних напрямків поточного розвитку Web-сервісів. Ключовим рішенням SWS може бути семантичне посередництво. Оскільки моделі сервісів і протоколи були розроблені кількома домінуючими організаціями, то існує певна кількість конкуруючих стандартів у цій галузі;

- **Grid** – Grid-системи для наукових обчислень призначені для розподілених великомасштабних гетерогенних обчислень. Ця область незалежно розвивається за стандартами Відкритої Архітектури Grid-сервісів. Місце SWS є домінуючим, що визначено розвитком напрямку e-science;

- **Ubiquitous** – Архітектура, яка спрямована на підтримку мобільного обчислення, що розвивається, з використанням портативних пристроїв, як то мобільні телефони, PocketPC та інших мобільних засобів. Місце SWS полягає у тому, щоб дати можливість використовувати в таких пристроях різноманітність сервісів, визначених за контекстом. Через надзвичайно динамічну зміну сервісів семантичні описи будуть мати важливе значення в таких архітектурах;

- **Web** – Ключовою мотивацією використання SWS в Internet є здатність динамічно знаходити і об'єднувати Web-сервіси під управлінням програмних агентів для таких використань, наприклад, порівняльні відвідини магазинів, координація спеціалізованих ділових послуг (наприклад, плани, подорожі, заходів, що передують певній зустрічі, конференцій та ін.) тощо;

- **Програмні Агенти** – Агент-орієнтовані технології з використанням SWS направлені на потреби великих гетерогенних систем, для забезпечення розподіленого планування й управління або координації, наприклад, у задачах логістики, де організаційна функція може мати певну перевагу, використовуючи методи SWS.

Існують проблемні, архітектурні та функціональні відмінності засобів SWS у перелічених архітектурах, але існують також подібності, дослідження яких пов'язано з дослідженням абстрактних архітектур SWS. Проблемними аспектами, які мають загальний характер і які визначені як семантичні функції Web-сервісів, є:

- ідентифікація сервісів (matchmaking) та їх пошук;
- автоматизована композиція сервісів;
- дослідження стану (моніторинг) процесів і сервісів;
- протоколи взаємодії сервісів, координація – інтерпретація і виконання;
- семантичне посередництво (наприклад, знаходження сервісів за змістом запиту або повідомлення, описів сервісу або заяв сервісів про своє існування);
- сервіс-орієнтоване планування (формування повідомлень або запитів) і інтерпретація відповідей;
- переговори і укладання контрактів;
- управління онтологіями, створення, пошук і доступ до онтологій;
- управління процесами створення сервісів (фабрики сервісів, інсталяція, переміщення);
- посередництво процесам і делегування функцій;
- послуги репутації (історія використання сервісів);
- безпека сервісів (включаючи ідентифікацію, встановлення аутентичності, делегування і дозвіл на основі політик);
- обробка відмов сервісів і компенсація.

До цих напрямків також відноситься розвиток методів використання сервісів у межах розвинутих сценаріїв на основі мультиагентних систем та співтовариства сервісів. Акцент у цьому напрямку спрямований на використання програмних агентів.

Розглянемо загальні аспекти, засоби та складові Семантичної мережі, на яких засновано підходи до рішення деяких з перелічених задач в SWS системах, з урахуванням нових концепцій представлення та обробки інформації.

2. Засоби представлення даних та знань в Семантичній мережі

Модель Семантичної мережі можна представити як сумісний розвиток двох напрямів, перший з яких охоплює мови представлення даних. На сьогоднішній день основними такими мовами є XML (eXtensible Markup Language) і засоби опису ресурсів RDF (Resource Description Framework), що мають статус рекомендацій W3C, та мова онтологій OWL, яка дозволяє визначити поняття і відношення між ними. Другий, концептуальний напрямок, несе в собі теоретичне уявлення про моделі предметних областей. Такі моделі предметних областей в термінології Семантичної мережі називаються онтологіями. В лютому 2004 року консорціумом W3C затвердив і опублікував специфікацію мови мережних онтологій OWL (Web Ontology Language) [6].

Таким чином, в Семантичній мережі використовують три ключові мови і відповідні технології:

- специфікація XML, що дозволяє визначити синтаксис і структуру документів;
- засіб опису ресурсів RDF, що забезпечує модель кодування для знань, визначених в онтології;
- мова онтологій OWL, що дозволяє визначити поняття і відношення між ними.

2.1. Онтології як засіб представлення знань. Мова онтологій OWL призначена для опису класів і відношень між ними, які властиві мережним документам (даним) і застосуванням (програмам). OWL забезпечує більш повну автоматичну обробку мережного контенту, ніж ту, яку підтримують XML і RDF, надаючи разом з формальною семантикою додаткову семантичну підтримку. При цьому самі онтології утворюють систему, що складається з наборів понять і тверджень, на основі яких можна будувати класи, об'єкти і відношення. Окрема онтологія визначає семантику конкретної предметної області й сприяє встановленню зв'язків між значеннями її елементів.

Онтології отримали досить широке розповсюдження в задачах представлення знань, інженерії знань, семантичної інтеграції інформаційних ресурсів, інформаційного пошуку і т.д. В штучному інтелекті онтологія визначається як „специфікація концептуалізації предметної області”, або спрощено, документ, що формально задає відношення між термінами. Це свого роду словник понять предметної області і сукупності явним чином виражених припущень щодо значення цих понять.

Розробка мови опису структурованих онтологій OWL стала однією з найважливіших робіт по Семантичній мережі, що проводяться консорціумом W3C. Для реалізації цієї мети у складі W3C створена спеціальна робоча група – Web Ontology Working Group. В лютому 2004 року WWW-консорціум надав мові OWL статус рекомендованої до реалізації технології [6].

OWL онтологія – це сукупність тверджень, які задають відношення між поняттями і визначають логічні правила для міркувань про них. Комп'ютери можуть „розуміти” значення даних на Web-сторінках, коли проходять послідовно по гіперпосиланнях до онтологічних ресурсів. Онтологія може включати описи класів, властивостей і їх екземпляри. Формальна семантика OWL описує, як отримати логічні висновки на основі онтологій, тобто отримати факти, які не представлені буквально, а виходять з семантики онтології. Ці висновки можуть базуватися на аналізі одного документа або множини документів, розподілених в мережі. Для створення онтології, яка може однозначно інтерпретуватися і використовуватися програмними агентами, застосовуються синтаксис та формальна семантика OWL.

Одна з проблем, що супроводжує створення та використання онтологій, полягає у тому, щоб змістовно поєднати єдиний смисловий простір. Ситуація аналогічна системам програмування на базі мов високого рівня. Мова дійсно дає універсальний засіб, проте надзвичайно важко досягти сумісності бібліотек, які спроектовані різними виробниками. Досвід показує, що потрібні стандартні бібліотеки.

Сучасна теорія не дає надійного способу поєднання даних, визначених різними онтологіями. Тому основним підходом до рішення цієї проблеми є наявність єдиної базової онтології щодо загального поля не специфікованої інформації і її розширень. Таку онтологію природно назвати „моделлю світу”, підкреслюючи її фундаментальну роль щодо моделей предметних областей.

Засобам представлення знань в Семантичній мережі властиві універсальні виразні можливості, синтаксична і семантична інтероперабельність. Мови Семантичної мережі дозволяють представляти будь-який вид даних, створювати численні синтаксичні аналізатори і інтерфейси прикладних програм, які дозволяють маніпулювати даними. Семантична інтероперабельність реалізується, наприклад, в онтологіях шляхом встановлення відповідності між термінами, що використовуються [7].

3. Рішення задач в системах нового покоління

Поряд з технологіями та засобами представлення даних та знань в Семантичній мережі народжується нова парадигма програмування, яка відповідає новим вимогам. Створення концепції процедурного, модульного і об'єктно-орієнтованого програмування – це приклади минулого, майбутнє, схоже, за сервісно-орієнтованою або агентно-орієнтованою архітектурою [8].

Архітектура застосувань Семантичної мережі складається з розподіленої системи інформаційних ресурсів (XML-документів, RDF-даних, OWL), опублікованих у мережі, інтерфейсів до даних, що дозволяють людям та комп'ютерам використовувати і змінювати їх у межах своїх повноважень. Програмні застосування складаються з мережних сервісів, що представляють собою стандартизовані прикладні програми, які апробовані і спрямовані на вирішення певної бізнес чи наукової задачі, а також універсальних сервісів та агентів, які вирішують задачі управління, забезпечуючи цілісність даних і коректність функціонування програмного середовища. Окремою складовою архітектури є засоби, що забезпечують пошук та доступ до сервісів мережі, запуск сервісів, композиції та моніторингу виконання сервісів [9].

Позитивною якістю ідей, закладених в основу Семантичної мережі, є гнучкість і відносна простота базових блоків, з яких може бути створена інформаційна система. Архітектура системи рішення задач в Семантичній мережі показана на рис. 2.

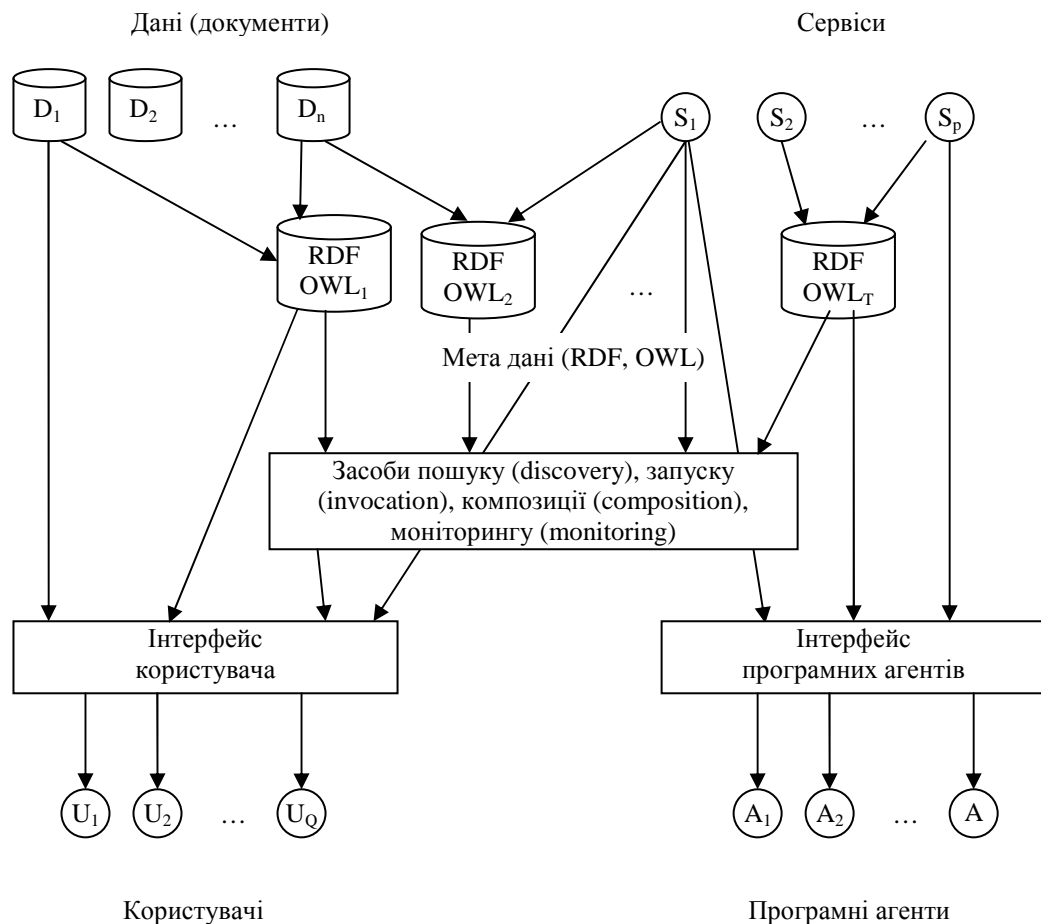


Рис. 2. Архітектура системи рішення задач в Семантичній мережі

Дані, метадані, інші структурні конструкції потрібні для цілісного опису інформаційної системи. Проблемними питаннями при побудові архітектури систем нового покоління можна назвати такі:

- прийняті технологічні рішення дозволяють об'єднувати дані з різних джерел, хоча єдиного підходу для об'єднання на семантичному рівні поки не створено;
- неопрацьованим є питання специфікації функціонування інформаційної системи і специфікації інтерфейсів.

3.1. Засоби сервісно-орієнтованої архітектури систем. Сервісно-орієнтована архітектура SOA (Service Oriented Architecture) визнана революційною в технології програмування. Насправді SOA – це всього лише інший стиль побудови сучасних корпоративних систем. Він орієнтується на сервіси і характеризується розподіленою архітектурою і слабо зв'язаними інтерфейсами [10].

Сервіс в поточному використанні – це не що інше, як одиниця робіт, що виконується сервіс-провайдером (службою, що надає сервіс) для забезпечення бажаного результату користувачеві сервісу. Саме сервіс, а не об'єкт, як в ООП, є тим модулем або програмним кодом, що повторно використовується, і при цьому він не залежить від технологій, мовних середовищ і інших ресурсів. Інтегруючу роль між сервіс-провайдером і користувачем беруть на себе програмні агенти.

За визначенням W3C, **Web-сервіс (Web service)** спрямований на забезпечення взаємодії „машина-машина” через мережу. Web-сервіс має інтерфейс, описаний у форматі, доступному для машинної обробки (WSDL) [11]. Інші сервіси також взаємодіють з Web-службою так, як це визначено їх описом, з використанням SOAP-повідомлень, що передаються по протоколу HTTP в поєднанні з іншими Web-стандартами.

Для взаємодії сервісів використовується порівняно невеликий набір простих інтерфейсів, які володіють тільки загальною семантикою і доступні всім провайдерам і користувачам. Через ці інтерфейси передаються повідомлення, які обмежені деяким словником. А оскільки використовується тільки загальна структура корпоративної системи і словник, то вся семантика і бізнес-логіка, специфічна для застосувань, описується безпосередньо в цих повідомленнях.

Корпоративна інформаційна система, побудована на основі SOA, складається з набору сервісів, доступних через прикладні програмні інтерфейси, та вбудованого механізму пошуку. Перелік сервісів у загальному реєстрі дозволяє користувачеві визначити службу та сервіс, що пропонує шукану функцію. Для вирішення певної задачі, користувач зв'язується з сервіс-провайдером і через прикладні програмні інтерфейси виконує необхідні дії. Ця процедура наочно показана на рис. 3.

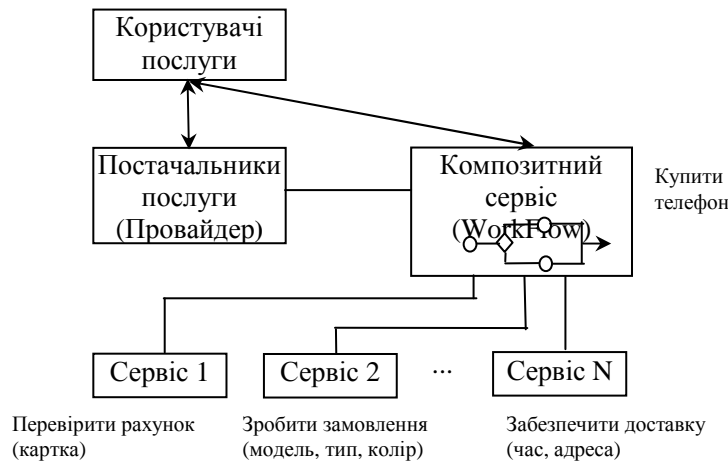


Рис. 3. Схема використання композитного сервісу

Всі компоненти корпоративної системи повинні підтримувати спільну мову, на якій формуються повідомлення.

Архітектура Web-сервісів також є сервісно-орієнтованою. Більш того, Web-сервіси – це суть SOA з двома суттєвими обмеженнями: інтерфейси базуються на протоколах (HTTP, FTP, SMTP, TCP), а всі повідомлення описуються у форматі XML. Детальні описи стандарту Web-сервісів і специфікацій SOA наводяться на сайтах консорціуму W3C. На рис. 4 показана система відкритих стандартів, що закладені в основу SOA архітектури [12].

Практичні аспекти сервісно-орієнтованої технології дозволяють розв'язати проблеми масштабованості, інтегрувати мережі передачі даних, спростити процедури проектування і управління мережами, а також створити інші розподілені застосування на засадах вже існуючих, прозоро взаємодіючи з ресурсами систем за допомогою прикладних програмних інтерфейсів і відкритих стандартів.

Web-сервіс вирішує певні задачі за допомогою одного або декількох програмних агентів, діючих від імені їх власника (провайдера). Інтерфейс, семантика і допоміжні метадані сервісу описуються набором документів у форматі XML. Типи і шаблони повідомлень, які передаються при взаємодії сервісів, визначені у відповідному інтерфейсі, а їх семантика визначена на моделі поведінки взаємодіючих сервісів.

Web-сервіси спрощують засоби IT-інфраструктури. Загальні реєстри сервісів доступні будь-яким зацікавленим клієнтам у всьому світі, надбудова (на відміну від вбудови) Web-сервісів над корпоративними системами не вимагає їх істотних змін, а загальнодоступні API (прикладні програмні інтерфейси) вирішують проблеми повторного використання коду і масштабованості його застосування.

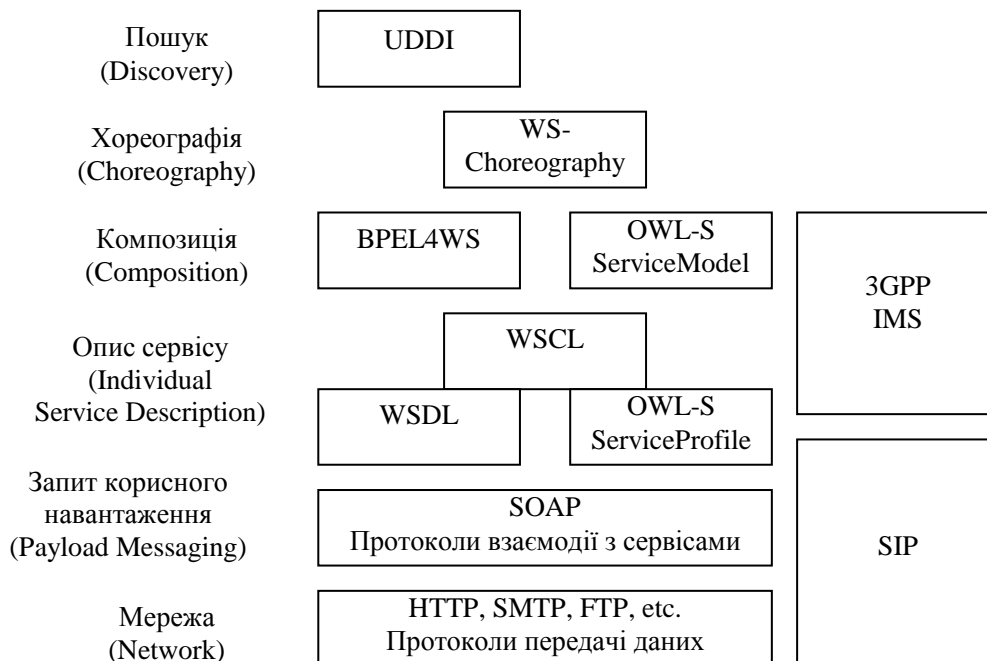


Рис. 4. Відкриті стандарти та засоби, що закладені в основу архітектури SOA

3.2. Семантичні WEB-сервіси. Які описуються з використанням мови семантичної розмітки. Такий семантичний опис дає змогу агентам „розуміти” SWS як за складом функцій, що вони виконують, так і за внутрішньою структурою, і тому агенти можуть знаходити, створювати композитний (складний) сервіс, запускати і контролювати такі сервіси.

Мовами семантичної розмітки SWS можуть бути DAML+OIL або OWL. Також ці мови комбінують з стандартними мовами опису Web-сервісів WSDL та SOAP. Хоч в SW-орієнтованих мовах програміст повинен створювати сервіси на концептуальному рівні для того, щоб уникати структурних помилок та невідповідностей, або помилок впродовж створення та розробки архітектури WS. В цьому сенсі розробка SWS потребує моделювання на концептуальному рівні і включає специфікацію опису сервісу, функціональні й структурні засоби.

SWS-композиція полягає у комбінуванні кількох сервісів для створення нового сервісу. Тому розробка SWS і композиція SWS мають схожі функції. Проте відмінність їх полягає у тому, що композиція використовує сервіси, які вже існують, і є напівавтоматичним процесом. При розробці SWS користувачі вручну створюють сервіси, використовуючи графічний інтерфейс.

Ми пропонуємо підхід до розробки композиції SWS, заснований на рівневій системі онтологій, які описують різні семантичні засоби та забезпечують можливості моделювати SWS з використанням PSM (Problem Solving Methods) методу, який описує та моделює внутрішню структуру сервісів [13].

3.3. Роль онтологій для розробки SWS. Відповідно до концептуальної архітектури для функціонування SWS і WS необхідні чотири типи засобів, що дають змогу зовнішнім програмам або агентам знаходити, запускати та створювати нові сервіси шляхом композиції вже існуючих:

засоби доступу (комунікації), що описують протоколи комунікацій, такі як SOAP або HTTP, необхідні для забезпечення запуску сервісів на виконання;

дескриптивні засоби SWS, що описують джерела, якими користуються функції сервісів. Це певні довідники та класифікатори, необхідні для виконання функцій WS. Вони можуть також допомагати в пошуці сервісів при відмові сервісу, який взаємодіє з різними джерелами;

функціональні засоби специфікують: потреби SWS щодо вхідних та вихідних даних, умови, які описують результати WS та вимоги до вхідних та вихідних даних, необхідних для виконання WS. Ці засоби дають змогу зовнішнім агентам визначати результат, отриманий при виконанні даного сервісу;

структурні засоби описують внутрішню структуру композиційного сервісу, яка складається з структурних компонентів (складових сервісів), і як ці компоненти пов'язані між собою при виконанні сервісу.

Агенти використовують ці засоби для композиції сервісів, так як вони визначають, чи існує зв'язок між складовою сервісу і другим сервісом, який використаний для композиції нового сервісу. Ці засоби представляють різні, але комплементарні аспекти сервісів, які необхідні агентам для виконання операцій (запуску, пошуку/публікації, композиції тощо). Наприклад, для запуску SWS агент повинен визначити як забезпечений доступ до сервісу, так і місце знаходження сервісу, вхідні та вихідні дані, проте йому не треба знати внутрішню структуру сервісу (як саме працює сервіс).

Ціль розробки SWS - створення детальних описів структурних компонент сервісів у зазначених напрямках, що гарантує коректність і уникає невідповідності складових сервісів, які створюються вручну або автоматично для забезпечення вимог сервісів та гарантованого їх виконання.

Традиційно розробники Web-сервісів для бізнес-процесів визначають множину дій для виконання процесів. Такий підхід розкладає процес (або сервіс) на активності, взаємодія яких може бути змодельована засобами потоку робіт (Work Flow - WF). Це головним чином визначає тільки координацію цих активностей у процесі їх виконання.

Заснований на онтологіях підхід до створення SWS використовує наступні типи онтологій:

онтології методу вирішення проблем (PSM), які використовуються для опису внутрішньої структури та функціональних засобів;

SWS онтології, що описують верхній рівень концептів у моделі;

онтології представлення знань, що визначають сутність предметної області SWS;

онтології типів даних, що описують типи даних, які використовуються в онтологіях.

3.4. Роль агентів в рішенні задач Семантичної мережі. В запропонованому підході маємо розрізнити мережні сервіси й агенти. Розглянемо основні відмінності між ними:

- мережні сервіси - WS безпосередньо «знають» тільки про себе, вони не володіють будь-яким мета-горизонтальним усвідомленням;

- мережні сервіси не розроблені для того, щоб використовувати або “розуміти” онтології;

- мережні сервіси не здатні до кооперативної дії через комунікації, або не здатні до кооперативної поведінки [10].

Напроти до вищезазначеного, агенти володіють всіма цими властивостями. Агент визначається як сервіс, який володіє такими основними властивостями як: ситуативність, автономність, реактивність, проактивність і соціальна здатність [8].

Метафора соціальної здатності надає особливій потужності агентно-орієнтованій парадигмі і є однією з особливостей, що робить агента спроможним для створення розподіленої системи. Якщо в мультиагентній системі кожен агент представляє індивідуальний сервіс, який в свою чергу, отримує можливість координуватися через ці агенти, то мережні сервіси також отримують соціальну властивість. Агенти

забезпечують динамічні соціальні форми поведінки сервісів, через які вони розділяють зобов'язання для досягнення загальної мети [8].

Використання методу потоку робіт (WF) в мультиагентній системі може розглядатися як засіб координації дій агентів для досягнення кооперативного рішення задачі [8]. Для цього кожен агент повинен визнати, що кращий шлях досягнення мети полягає у залученні допомоги інших агентів. Суспільні зобов'язання виникають відповідно до суспільної залежності, коли один агент робить зобов'язання щодо іншого.

Суспільні агенти, які мають доступ до семантичного опису (онтології), спроможні до координування своєї поведінки. Навіть без семантичної інформації, агенти можуть використовуватись для виконання локальної оптимізації потоку робіт. Агенти використовують семантичні описи поведінки для того, щоб обдумати свої власні можливості й можливості інших агентів у системі, й використати ці міркування для кооперативного вирішення задачі.

Ми пропонуємо новий підхід, який базується на використанні опису обчислювального процесу для того, щоб нав'язати певний порядок у поведінці колекції агентів та відповідних їм сервісів.

У технологічному процесі рішення задач агенти представляють мережу сервісів, за умови, що їх відносини апріорі відомі. Наша стратегія полягає у тому, щоб створити модель або обчислювальну граф-схему, відповідну умовам та специфікації задачі, для керування та координації дій агентів в технологічному процесі рішення задачі.

3.5. Модель потоку робіт як засіб мультиагентної інтеграції. Модель потоку робіт представляється як корисний інструмент моделювання технологічного процесу, в якому однозначна і точна семантика дозволяє представити поведінку технологічного процесу, яка надалі буде проаналізована агентами. В результаті аналізу можна визначити статичні та динамічні властивості технологічного процесу, включаючи відсутність або наявність критичних ділянок, тупикових станів тощо. Отримана інформація дозволить агентам ефективно координувати свої дії при виконанні технологічного процесу рішення задачі.

Архітектура системи рішення задач з використанням мультиагентного потоку робіт показана на рис. 5.

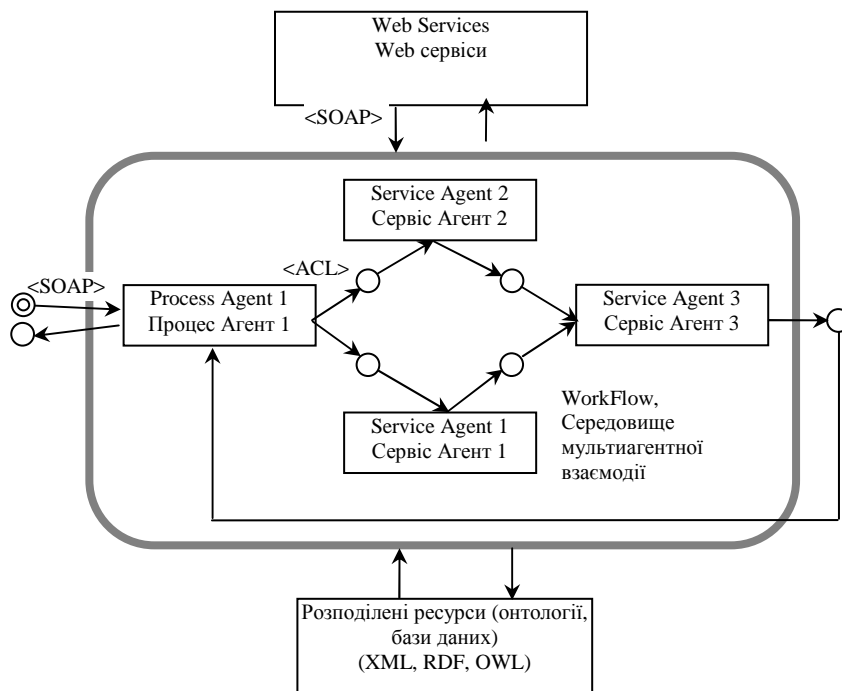


Рис. 5. Архітектура мультиагентної координації сервісів з використанням Work Flow засобів

Використання мультиагентної системи для керування технологічним процесом рішення задач є кроком, який потребує дослідження багатьох фундаментальних питань.

Як може окремих агент діяти в мультиагентному оточенні та використовувати WF засоби? Засоби WF в мультиагентній реалізації можуть забезпечити моніторинг дії агентів та системи в цілому. Без цього процес виконання агентів є в більшому ступені автономним. З іншої сторони, агент, який моніториться, взаємодіє з учасниками процесу та повинен знаходити в WF-сценарії об'єкт взаємодії, формувати правила взаємодії та використовувати їх в умовах, коли він працює автономно. Причому інформація, яку агент отримує з Workflow, повинна дозволити йому діяти без додаткового ручного управління (самоорганізація).

Як можуть агенти посилити можливості Workflow-засобів, які можуть бути використані в бізнес-логіці для пошуку, відбору і послідовності виконання бізнес-сервісів? Метаінформація сервісів далі може бути використана автономними агентами для перебудови Workflow (самооптимізація). Наприклад, агенти можуть використовувати Workflow-опис для визначення можливих змін в технологічному процесі, щоб використати

цю інформацію разом зі знаннями про доступні ресурси для пошуку інших ресурсів, які були б необхідні для рішення задачі на наступних кроках технологічного процесу.

Як необхідно розширити засоби опису та специфікації Workflow, зокрема BPEL4WS, для того щоб забезпечити специфікацію множини функціональних рівнів? Наприклад, для створення Workflow по доставці товарів можна використовувати інформацію про ціни, час, якість послуг тощо (самоконфігурація). Більше того, список партнерів може бути представлений попередньо і повторно провайдером сервісів. У випадку невдалого попереднього представлення WF може автоматично визначити наступного партнера (самовідновлення).

Як може моніторинг активних агентів визначати або корегувати помилки доступу до сервісу (самозахист)? Якщо агент виконує ідентифікацію функції нормальної поведінки системи і в разі відхилення повинен виставити повідомлення, що все що відбувається – ненормально. Ненормальна поведінка може бути потім проаналізована для визначення реальної загрози або відхилення, що викликане діями агентів.

3.6. Моделі та засоби координації мережних сервісів на основі потоку робіт. Ключовим значенням для вирішення проблеми інтеграції сервісів є забезпечення стандарту взаємодії сервісів або виконання бізнес-процесів. Для вирішення цієї задачі запропоновано засоби BPEL (Business Process Execution Language). Засоби BPEL є новим стандартом для інтеграції гетерогенних застосувань і сервісів, і є мовою потоків робіт, процесів і даних. Основу BPEL складають три ключові властивості: асинхронність, координація потоків і управління винятковими ситуаціями [13-15].

Стандарти управління потоками робіт забезпечують ряд очевидних таких переваг:

- строга формалізація процесу опису потоків робіт, що забезпечує стандартний процес аналізу і побудови нових моделей складних робочих процесів;
- переносимість і інтероперабельність: моделі процесів, створені в межах однієї системи, можуть частково або повністю працювати під управлінням іншої системи;
- універсальність: розширюваність стандартів робить можливим застосування єдиного механізму опису управління потоками робіт у різних сферах діяльності.

Специфікація кожного потоку робіт представляється набором взаємодіючих між собою компонентів (рис. 6).

На найвищому рівні всі WF системи можуть характеризуватися як системи, що забезпечують підтримку управління сервісами в трьох функціональних областях:

- вбудовані функції, які задаються визначенням і моделюванням процесу WF (Build-time functions);
- функції контролю, що контролюють управління процесами під час виконання WF (Run-time control functions);
- взаємодії з користувачами під час виконання процесу (Run-time Activity Interactions).

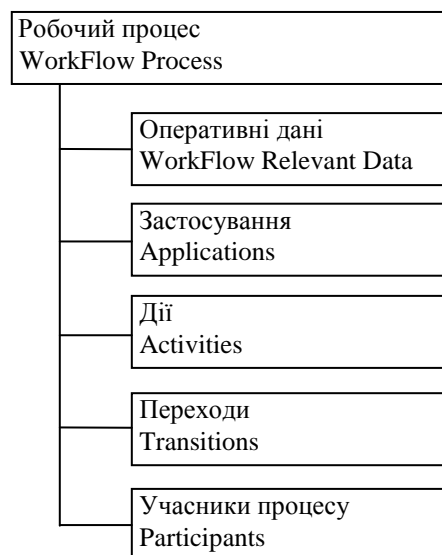


Рис. 6. Схема компонентів специфікації потоку робіт

Основні характеристики Workflow-систем можна визначити наступними функціональними рівнями архітектури управління потоками робіт [15]:

Рівень визначення процесу. Даний рівень представлений набором функцій, задача яких полягає у декларативному визначенні потоку робіт бізнес-процесу. Під час цієї фази бізнес-процес переводиться з „реального світу” в „формальний”, який розуміється комп'ютером, за допомогою аналізу, моделювання і використання набору засобів декларативного представлення. Результуюча декларація процесу носить назву „моделі процесу”, „шаблона процесу”, „метаданих процесу” або „визначення процесу”. Модель процесу

складається з ряду дискретних „дій”, перехід між якими здійснюється відповідно до набору деяких правил за допомогою виконання операцій, виконавцем яких може бути як людина, так і комп'ютер. Визначення процесу може бути представлено як текстовим, так і графічним описом, або декларацією на деякій формальній мові опису потоків робіт. Деякі процеси допускають можливість внесення змін в їх визначення в ході свого виконання.

Рівень створення і виконання процесу. На етапі виконання відбувається аналіз визначення процесу спеціальним програмним забезпеченням, яке відповідає за створення і управління екземплярами потоків робіт, управління і планування завдань, залучення відповідних людських, мережних ресурсів. Функції рівня виконання є зв'язуючою ланкою між моделлю процесу в формальному визначенні і його реальним станом, що відображає взаємодію користувачів і застосувань. Центральним компонентом системи є служба управління потоками робіт, яка відповідає за створення і видалення процесів, планування завдань процесу і взаємодію з людськими і автоматизованими ресурсами.

Рівень взаємодії з користувачами і застосуваннями. Індивідуальні завдання робочого процесу, як правило, пов'язані з людською діяльністю, або станом певного ресурсу в системі, наприклад, занесення нового запису в базу даних. Тому необхідно забезпечити гнучку координацію користувачів із службою управління потоками робіт для того, щоб передавати управління між завданнями, призначати операційні стани процесам, запускати застосування або передавати їм необхідні дані.

Виділяють наступні функціональні модулі WF систем:

- визначення процесу робіт: специфікація даних сервісу та його взаємодія з навколишнім середовищем;
- підтримки інтероперабельності між різними системами WF;
- виклик сторонніх застосувань;
- підтримки інтерфейсу взаємодії з призначеними для користувача функціями (наприклад, обробка запиту на отримання нового завдання або завершення того, що виконується);
- підтримки інтерфейсів для забезпечення контролю системи, а також для полегшення управління взаємодією з навколишнім середовищем, адміністрування і контроль.

Заклучення

У даній роботі були представлені основні стандарти та засоби, що закладені в основу технології сервіс-орієнтованої інтеграції, спрямованої на проблематику рішення задач в середовищі Семантичної мережі. Семантичні технології забезпечують існування певного рівня абстракції над існуючими інформаційними технологіями. Цей рівень дозволяє здійснювати зв'язок даних, семантики і процесів, що виконуються в розподіленому середовищі.

Розвиток цих технологій дасть змогу створювати єдине уніфіковане представлення даних у всіх застосуваннях, дозволить точно знаходити необхідну інформацію, спрощувати корпоративну інтеграцію і інтеграцію мережних застосувань. Запропонована схема для представлення та рішення задач у розподіленому середовищі Семантичної мережі.

Розглянуто засоби та стандарти інтеграції та координації мережних сервісів на основі потоку робіт.

1. *Tim Berners-Lee, James Hendler and Ora Lassila.* The Semantic Web <http://www.sciam.com/article.cfm?articleid=000A0919>
2. <http://www.semwebcentral.org>
3. *Sheila A. McIlraith, Tran Cao Son, and Honglei Zen.* Semantic Web Services, Stanford University. <http://www.ksl.stanford.edu>.
4. *Web Services.* <http://www-306.ibm.com/software/solutions/webservices/uddi>
5. <http://www.webservices.org/>, <http://www.w3.org/TR>
6. *OWL Technical Committee.* Web Ontology Language (OWL). <http://www.w3.org/TR/2004/WD-owlref>
7. *Калиниченко Л. А.* Методология организации решения задач над множественными распределенными неоднородными источниками информации. ИПИ РАН, http://www.rfbr.ru/default.asp?doc_id=20786
8. *Wooldridge M.* - On the Sources of Complexity in Agent Design - In Applied Artificial Intelligence, 2000. - 14(7), P. 623 - 644,
9. *Andon Ph., Deretsky V.* Control Oriented Ontology and Process Description for Cooperation Agents in Information Retrieval// Sixth International Scientific Conference „Electronic Computers and Informatics ECI'2004”. – Kosice – Herlany, Slovakia September 22-24, 2004. - P. 14 – 18.
10. *Hull R.* Web Services Architecture, W3C Working Group Note 11, February, W3C Technical Reports and Publications, <http://www.w3.org/TR/ws-arch/>, <http://www.informatik.uni-trier.de/~ley/db/conf/icws/icws2005.html>
11. <http://developers.sun.com/techtoc/webservices/wscf/wscf.pdf>
12. *A Comparison of XPD, BPML and BPML4WS* // <http://xml.coverpages.org/Shapiro-XPDL.pdf>
13. *Asunción Gómez-Pérez and Rafael González-Caber.* ODE SWS: A Framework for Designing and Composing Semantic Web Services Technical University of Madrid. www.delicias.dia.fi.upm.es.
14. *Workflow Management Coalition standards* // <http://www.wfmc.org/standards/standards.htm>
15. *Workflow Process Definitio*, http://www.wfmc.org/standards/docs/TC-1025_10_xpdl_102502.pdf