

УДК 004.32

І. О. Храмова

Інститут проблем реєстрації інформації НАН України
вул. М. Шпака, 2, 03113 Київ, Україна

Застосування сервісно-орієнтованих архітектур у процесах інтеграції інформаційних ресурсів

Розглянуто окремі аспекти застосування сервісно-орієнтованої архітектури для рішення завдань горизонтальної інтеграції автономних інформаційних ресурсів.

Ключові слова: *сервісно-орієнтована архітектура, інформаційний ресурс, інтеграція, метадані.*

Сучасне велике підприємство важко уявити без інформаційних систем різного призначення: бухгалтерських, фінансово-аналітичних, планово-виробничих, складських. При цьому мова йде про складні багатофункціональні інформаційні системи (ERP, CRM, SCM тощо). Часто в експлуатації знаходиться декілька таких систем одночасно. Якщо врахувати зовнішні, не менш складні і специфічні, інформаційні системи, з якими потрібно обмінюватися тією чи іншою інформацією, виникає закономірне питання: яким чином організувати взаємодію всіх цих систем? Не менш життєвим питанням буде й те, як організувати цю взаємодію *ефективно*, щоб створити продуктивні, надійні та безпечні автоматизовані наскрізні міжорганізаційні (тобто такі, що простираються за межі підприємства) низки тих ділових процесів, виконання яких необхідне підприємству для здійснення своєї діяльності? Саме тут найбільш ефективним інструментом рішення виступають WEB-сервіси (далі — сервіси) як набір стандартів і специфікацій для спільної роботи і взаємодії різних ділових об'єктів між собою з метою виконання визначеної функції у рамках здійснюваної виробничої діяльності.

Архітектура застосувань, у рамках якої всі функції застосування є незалежними сервісами з чітко визначеними інтерфейсами, які можна викликати в потрібній послідовності з метою формування бізнес-процесів, отримала назву сервісно-орієнтованої архітектури (SOA)¹.

Виходячи з цього визначення, якщо будь-яке застосування з функціональної точки зору може бути представленим сукупністю функцій, то ресурс, що реалізує

© І. О. Храмова

¹ На сьогодні стійкого, прийнятого всім ІТ-суспільством, визначення SOA немає. Приведене визначення, на погляд автора, найбільш повно та точно відображає сучасний стан цієї концепції.

функцію надання, перетворення чи завантаження даних, є сервіс. Таким чином, наведене визначення потребує для представлення та реалізації будь-якого застосування в рамках SOA проведення його повної декомпозиції до рівня окремих функцій маніпулювання даними.

У поняття незалежності сервісу вкладається наступний зміст: сервіси функціонують незалежно від інформаційних систем, є функціонально самостійними об'єктами. Для будь-яких зовнішніх застосувань вони виступають як «чорні ящики»: зовнішні застосування не знають, як сервіс отримує дані. Застосуванню відомо тільки, які дані необхідно подати на вхід сервісу і що потрібно очікувати на його виході.

Чітко визначені інтерфейси є запорукою успішного застосування сервісів. Це означає, що функція (або функції), яку реалізує даний сервіс, повинна бути однозначно описана згідно визначених правил, прийнятих для всіх сервісів. Повинні бути описані як набір і типи вхідних даних, так і набір і типи вихідних даних.

При цьому, для зовнішніх по відношенню до сервісу інформаційних систем мова програмування, за допомогою якої реалізовано цей сервіс, програмно-апаратна платформа, на якій він функціонує, не повинні мати значення. Також, не повинно мати значення, є даний сервіс локальним, чи віддаленим. Зовнішня інформаційна система повинна мати можливість взаємодіяти із сервісом (тобто передати йому вхідні дані та одержати вихідні) поза залежністю від указаних особливостей.

Основні компоненти та протоколи

Основні компоненти сучасних сервісів можна приблизно класифікувати наступним чином.

Споживачі сервісів. Споживачами сервісів є кінцеві користувачі та пристрої.

Інтерфейс сервісу. Інтерфейс сервісу реалізує програмний інтерфейс для з'єднання та передачі повідомлень. Реалізує протоколи, перетворює та моделює дані для потрібного пристрою, а також об'єднує, персоналізує, реєструє та локалізує вміст.

Контейнер сервісу. Контейнер сервісу реалізує програмне оточення, функції терміну життя (персистентності) та здійснює управління станом.

Платформа сервісу. Платформою сервісу слугує система або обладнання, на яких він працює. У функції платформи входить забезпечення доступу до баз даних, служби обміну повідомленнями, служби каталогу, віртуальних машин, операційних систем, апаратних інтерфейсів і пам'яті.

Бізнес-логіка. Бізнес-логіка — це програмний код, що реалізує ділові процеси.

Інтеграція сервісу. До функцій інтеграції сервісу відносяться: об'єднання інших сервісів, діючих систем, а також служб інфраструктури, таких як бази даних, файли та директорії.

Сучасна модель сервісів повинна надавати можливість швидкої комбінації та рекомбінації компонент. Це досягається такими інструментами як:

відкриття — відкриває інші служби за допомогою реєстраційних форм та репозиторіїв ebXML та/або UDDI [1, 2];

створення — створює XML-контент і формує бізнес-логіку;

перетворення — перетворює дані у формат застосування XML, Java або SQL;
конструювання — розміщує компоненти в інтерфейс XML-повідомлень, використовуючи службу пересилання повідомлень ebXML або SOAP [3];
розгортання — передає сервіс механізму розміщення сервісів;
тестування — локально або віддалено тестує застосування;
публікація — розміщує сервіси в реєстрах типу ebXML або UDDI.

Проблеми інтеоперабельності інформаційних ресурсів

Кожен розробник створює для своєї прикладної системи власну семантику, іноді про те, що конкретно в системі означає певне поле, не знає навіть її розробник. XML [4] спочатку розглядали як одну зі спроб вирішити цю проблему. Фактично, XML дозволяє отримати програмним посередникам (брокерам), які здійснюють семантичні перетворення, інформацію про структуру даних, але не більше того. Отже, незважаючи на зростання можливостей формалізувати семантику сервісів, проблема семантичного перетворення ще існує. І брокер повідомлень, що обробляє інформацію та перетворює її у відповідності з семантикою, описом і внутрішніми стандартами, в тому чи іншому вигляді стає обов'язковим компонентом SOA. WEB-сервіси дуже динамічно удосконалюються в напрямку вирішення цієї проблеми. З однієї сторони, є намагання полегшити доступ за рахунок інфраструктури, а з іншої — використати XML як засіб, що полегшує розв'язання семантичних проблем.

Інтеграція є багаторівневим процесом. На рівні бази даних застосування обмінюються інформацією. На наступному рівні вже здійснюється об'єктна взаємодія. Потрібно знати, який об'єкт треба викликати для того, щоб запустити певний діловий процес в іншій системі. Далі йде рівень взаємодії за допомогою документів і понять предметної галузі. На цьому рівні оминають технологічну складову, не формулюють запитів до бази даних, а просто оперують певними поняттями предметної галузі на взірць «замовлення» або «накладна».

Звичайно, можна інтегрувати інформаційні ресурси (IP) на рівні баз даних, але, коли мова йде про різні застосування та різні організаційні структури, інтеграція на рівні даних є надто обтяжливою. Спроби інтеграції на рівні об'єктів (CORBA), як виявилось, теж багато чого не дозволили реалізувати. Зараз відбувається етап стандартизації, що має підготувати технології інтеграції даних до реалізації на базі SOA. З'явилися поняття «сервіс даних»² та «дані-як-сервіс» (data-as-a-service), які об'єднуються в окремий рівень, що абстрагує бізнес-логіку від технології доставки даних із різних джерел споживачу (як застосуванням, так і користувачам) і технологій перетворення даних [5].

Найбільш вагомим проблемою створення сервісно-орієнтованої архітектури інтеграції IP треба вважати семантичне узгодження даних. Іншою, на сьогодні мало вивченою проблемою, є управління періодом життя (persistence) сервісів із метою забезпечення живучості сталих ділових процедур, реалізованих на засадах SOA.

² Концепція надання даних в якості сервісів лягла в основу концепції ІВМ «інформація за вимогою» (Information on Demand).

Рішення першої з них передбачає створення, так би мовити, семантичного опису (анотації) сервісу, який є екземпляром певного об'єкту предметної онтології, у якого чітко визначені структура, семантика і обмеження застосування [6, 7].

Рішення другої проблеми — у своєрідному резервуванні сервісів, тобто створенні та підтримці для кожного із задіяних сервісів переліку альтернативних сервісів, що зможуть у випадку програмної або апаратної відмови замінити основний.

Невирішеною залишається проблема, яка полягає у відносній незрілості (інколи, відсутності) і несумісності корпоративних стандартів щодо управління метаданими, в тому числі семантичними, окремих власників ІР. Розв'язання цієї проблеми є сферою піклування нормативного регулювання в галузі інформатизації державного управління.

Аналіз ролі метаданих і підходів до їхнього застосування в процесі інтеграції інформаційних ресурсів

Зазвичай під метаданими розуміється будь-яка інформація, необхідна в інформаційних технологіях для аналізування, проектування, побудови, впровадження і застосування комп'ютерної системи. У випадку інтеграції інформаційних систем метадані особливо спрощують управління, створення запитів, повноцінне використання та розуміння даних. Багато недавніх проектів, як наукових, так і практичних, було направлено на вивчення метаданих.

Створення метаданих і управління ними служить двом цілям [8]:

1) спрощенню робіт зі створення та управління інформаційною інфраструктурою;

2) більш ефективному витяганню інформації.

Перша з них, в основному, стосується кола таких аспектів управління інформаційною інфраструктурою як:

— підтримка інтеграції систем;

— підтримка аналізування та проектування нових застосувань;

— підвищення гнучкості системи та можливості неодноразового використання існуючих програмних модулів;

— автоматизація адміністративних процесів;

— посилення механізмів безпеки.

Друга ціль відноситься скоріше до кола питань покращання інформаційного сервісу, а саме:

— підвищення якості отримуваних даних;

— спрощення взаємодії всередині інформаційної інфраструктури;

— застосування загальної термінології та мови взаємодії.

Розглянемо викладене трохи докладніше.

Підтримка інтеграції систем. Схеми та інтеграція даних залежать від метаданих, які описують структуру та сенс окремих джерел даних і цільових систем. Правила перетворення можна застосувати до початкових даних і зберігати в якості метаданих. Більш того, інтеграція різних інструментів можлива тільки тоді, коли вони розділяють «дані», які в даному випадку представляють собою метадані інформаційного ресурсу.

Підтримка аналізування та проектування нових застосувань. Метадані підвищують можливість управління та надійність процесу розробки застосувань, забезпечуючи інформацію про суть даних, їхню структуру та джерела. Більш того, метадані, які стосуються рішень по проектуванню застосувань, можна використовувати неодноразово.

Підвищення гнучкості системи та можливості неодноразового використання існуючих програмних модулів стає можливим тільки для активного та напівактивного використання метаданих. Семантичні аспекти, що швидко змінюються, явним чином зберігаються у вигляді метаданих поза прикладними програмами. Тому підтримка істотно спрощується. Систему можна розширити та адаптувати без всяких труднощів. До того ж є можливість неодноразового використання окремих фрагментів програмного коду.

Автоматизація адміністративних процесів. Метадані управляють запуском на виконання різних процесів в інформаційному ресурсі (наприклад, завантаження та поновлення). Інформація про їхнє використання (журнали доступу, кількість доданих записів і т.п.) також знаходиться в репозиторії, який легко може бути використаний адміністратором.

Посилення механізмів безпеки. Метадані можуть і повинні забезпечити правила доступу та користувацькі права для всієї інформаційної інфраструктури. Управління доступом у джерелах даних іноді вимагає застосування складних методів. Наприклад, оперативне джерело може містити інформацію про окремі показники роботи підприємства, але агреговані в сховищі даних (СД) значення можуть складати таємницю. Зворотній приклад: персональні доходи кожного співробітника є конфіденційною інформацією, а загальний фінансовий показник зарплати, який зберігається в СД, може бути відкритою інформацією.

Підвищення якості отримуваних даних. Якість даних визначається наступними характеристиками:

- 1) узгодженістю (чи є надання даних однорідним, чи немає дублікатів, даних із перехресними або конфліктуючими визначеннями);
- 2) повнотою (чи всі дані присутні);
- 3) точністю (збігом фактичних значень і значень, що зберігаються);
- 4) своєчасністю (чи актуальне значення, що зберігається).

Правила перевірки якості даних необхідно задати, зберегти у вигляді метаданих і перевіряти дані при кожному поновленні інформаційного ресурсу. Крім того, високої якості потребує підтримка контролю даних. Метадані забезпечують інформацію про час створення, про автора даних, про джерело, значення даних у момент одержання (про наслідування даних), та про подальший шлях від джерела до поточного місцезнаходження. Таким чином, користувачі можуть відновити низку кроків обробки, по якій проходять дані за час перетворення, та перевірити точність поверненої інформації.

Спрощення взаємодії всередині інформаційної інфраструктури. Взаємодія відбувається як за допомогою виконання простих запитів і звітних застосувань, так і з використанням складних аналітичних інструментів. Метадані забезпечують відомості про значення даних, термінології і напрямках діяльності підприємства, а також їхній зв'язок з даними. Тому метадані підвищують якість запитів, що виконуються, за рахунок більш точного та суворого формулювання, а також скорочу-

ють витрати на користувачів, яким необхідні доступ, оцінка та застосування відповідної інформації.

Покращання аналізування даних. Методи аналізування даних наразі представлені достатньо широко: від простих застосувань звітності та OLAP³ до складних застосувань Data Mining and Knowledge Discovery⁴. Тут метадані необхідні для розуміння предметної області та її представлення в інформаційному ресурсі, з тим щоб надалі адекватно застосувати та інтерпретувати результати.

Застосування загальної термінології та мови взаємодії. Доступність метаданих як унікального джерела інформації для користувачів має й інші переваги. Вона гарантує узгоджені засоби взаємодії та інтерпретації інформації з IP, а також усуває двозначність та забезпечує узгодженість корпоративних відомостей, дозволяє розділяти знання та досвід.

З викладеного помітно, що всередині загальної інформаційної інфраструктури складається певне технологічне середовище для метаданих, яке відіграє центральну роль у сервісно-орієнтованій системі інтеграції даних. Зібрані в єдиному корпоративному репозиторії, метадані дозволяють задати спільну мову визначення даних і досягти єдиного розуміння даних на різних рівнях їхнього використання.

Репозиторій метаданих є структурованою системою зберігання та вибірки даних, яка реалізується на основі СУБД. Він включає визначення даних, правила перетворення в ETL⁵-системах, відображення з джерела даних у цільове сховище, логічні моделі даних та інші характеристики, що описують як шукати, здійснювати доступ і використовувати дані. Репозиторій дозволяє задати єдину семантику і правила до всіх ділових процесів, відстежувати і управляти даними і, подібно до універсального брокера, зв'язує високорівневі сервіси даних і детальніші визначення та залежності даних. Основна цінність такого репозиторію — в абстрагуванні логічних моделей даних від фактичної реалізації технологій інтеграції даних.

Семантичні сервіси метаданих дозволяють погоджувати семантику даних між різними системами, виявляти походження та історію даних для звітності і аудиту, відстежувати факт внесення змін, виявляти пов'язані з даними слабкі місця в загальній інформаційній архітектурі. В цілому середовище метаданих покликане забезпечити належну структуру та якість даних, перш ніж вони будуть передані відповідному бізнес-сервісу.

Наскрізне управління метаданими та підтримка зв'язків між джерелом і споживачем даних нерозривно зв'язане зі створенням та підтримкою сукупності стандартизованих понять предметної галузі, яка створюється у вигляді тезаурусу — ієрархічного словника понять із визначеннями, або більш складної структури —

³ OLAP (Online Analytical Processing) — оперативна аналітична обробка (даних).

⁴ Data Mining and Knowledge Discovery — міждисциплінарний напрямок, що фокусується на методологіях для виявлення (Discovery) та витягання (Mining) корисного знання (Knowledge) з даних (Data).

⁵ ETL (extraction, transformation and loading) пов'язується з поняттям сховища даних і асоціюється з пакетним виконанням операцій витягання, перетворення та завантаження даних.

онтології, що вміщує додатково зв'язки між поняттями та правила і обмеження у використанні понять.

Висновки

Застосування сервісно-орієнтованих архітектур у процесах інтеграції ІР є перспективним і багатообіцяючим напрямком створення інформаційних інфраструктур забезпечення наскрізної міжвідомчої інформаційної взаємодії. Проте застосування інформаційних сервісів потребує значних попередніх зусиль, пов'язаних із розв'язанням викладених у статті проблемних аспектів, найважливішими з яких є семантичне узгодження даних, управління періодом життя (persistence) сервісів, відсутність або незрілість і несумісність корпоративних стандартів щодо управління метаданими, в тому числі семантичними, окремих власників ІР.

Центральну роль у сервісно-орієнтованій системі інтеграції ІР відіграє середовище метаданих, що базується на єдиному корпоративному репозиторії, де збираються метадані ІР. Отримати «спільну мову» для визначення та осмислення даних і єдиного розуміння даних на різних рівнях їхнього використання можна шляхом створення та підтримки системи стандартизованих понять предметної галузі, яка реалізується у формі корпоративного тезаурусу або онтології.

Узгодження понять між окремими системами виконується семантичними сервісами спеціалізованої інфраструктури управління інтеграцією даних автономних ІР, яка покликана забезпечити належну структуру і якість даних, перш ніж вони будуть передані відповідному сервісу.

1. ebXML Specs. — Режим доступу: <http://www.ebxml.org/specs/index.htm>
2. UDDI Version 3.0.2 Specification. — Режим доступу: <http://www.oasis-open.org/committees/uddi-spec/doc/spec/v3/uddi-v3.0.2-20041019.htm>
3. SOAP Tutorial. — Режим доступу: <http://www.w3schools.com/soap/default.asp>
4. XML Tutorial. — Режим доступу: <http://www.w3schools.com/xml/default.asp>
5. Черняк Л. SOA и сервисы данных / Л. Черняк // Открытые системы. — 2008. — № 2. — Режим доступу: <http://www.osp.ru/os/2008/02/4923630/>
6. Матов О.Я. Проблемы горизонтальной интеграции информационных ресурсов в багаторівневих організаційних структурах з динамічною конфігурацією / О.Я Матов, І.О. Храмова // Реєстрація, зберігання і оброб. даних. — 2007. — Т.9, № 3. — С. 88–97.
7. Матов О.Я. Динамічна інтеграція інформаційних ресурсів єдиної інформаційної інфраструктури ринку електроенергії / О.Я Матов, І.О. Храмова // Функціонування та розвиток ринків електроенергії та газу: зб. наук. праць / Інститут проблем моделювання в енергетиці НАН України ім. Г.Є.Пухова. — К., 2006. — С. 93–98.
8. Матов О.Я. Перспективні інформаційні технології та розвиток GRID-систем у високопродуктивних глобально-розподілених обчислювальних інфраструктурах корпоративної співпраці / О.Я Матов, І.О. Храмова // Реєстрація, зберігання і оброб. даних. — 2004. — Т. 6, № 1. — С. 85–98.

Надійшла до редакції 12.02.2009