

Д.І. Угрин, В.В. Литвин  
 Буковинський університет, м.Чернівці,  
 Національний університет “Львівська політехніка”,  
 кафедра інформаційних систем та мереж

## ДОСЛІДЖЕННЯ СЕРВІСНО-ОРІЄНТОВАНОЇ АРХІТЕКТУРИ ЯК ІНТЕГРАЦІЙНОЇ ПЛАТФОРМИ В ТУРИСТИЧНІЙ СФЕРІ

© Угрин Д.І., Литвин В.В., 2011

Визначено особливості методики сервісно-орієнтованої архітектури. Проаналізовано на адекватність застосування механізму SOA для інформаційних систем туристичних агентств та фірм. Побудовано схеми створення автоматизованої системи з використанням уніфікованого процесу Rational Unified Process для туристичної сфери та еволюцію інформаційної системи на основі SOA в туристичній сфері з включенням успадкованого програмного забезпечення.

**Ключові слова:** сервісно-орієнтована архітектура, інформаційна система, бізнес-процес, інтеграція даних, туристична сфера.

**The features of methodology of the service-oriented architecture are certain. It is analysed on adequacy of application of mechanism of SOA for the informative systems of travel agencies and firms. The charts of creation of CAS with the use of compatible process of Rational Unified Process for a tourist sphere and evolution of the informative system are built on the basis of SOA in a tourist sphere with including of the inherited software.**

**Key words:** service-oriented architecture, informative system, business process, integration of data, tourist sphere.

### Постановка проблеми у загальному вигляді

Постійні зміни розвитку туристичного ринку та бізнес-процеси ставлять перед інформаційними службами туристичних організацій складне завдання – за допомогою засобів автоматизації максимально скоротити час реакції бізнесу туристичних організацій на будь-які зовнішні чи внутрішні зміни. Скорочення часу освоєння нових туристичних послуг, впровадження нових процесів туристичного напрямку та інших інновацій забезпечує туристичним організаціям, зокрема туристичним агентствам, підтримку конкурентоспроможності на внутрішньому і зовнішньому ринках.

Проте типова обчислювальна інфраструктура туристичних організацій характеризується фактичною ізоляваністю складових її застосувань, що з погляду бізнесу веде до затримок у виконанні бізнес-процесів, порушення взаємодії між підрозділами та партнерами, перешкоджає управлінню і контролю, призводячи до зниження ефективності роботи організації загалом. Відсутність інтегрованості ставить під питання корисність інформаційних технологій для туристичної сфери. Здебільшого ці проблеми виникають через відсутність архітектурних стандартів в області інформаційних технологій. Більше того, для їх подолання необхідні не лише стандарти, але й набір компонентів, які є стандартизованими. Настільки ж необхідне і середовище виконання бізнес-процесів, за допомогою якого компоненти будуть пов'язаними.

До кінця минулого століття під час побудови інформаційних систем домінував підхід, що ґрунтується на аналізі і управлінні інформаційними потоками. Проте з часом як універсальніший та інформативніший став розглядатися орієнтований в процесі підхід, в межах якого вибір і зміна структури інформаційної системи, склад її компонент та основні характеристики здійснюються від процесу залежно від його побудови та особливостей, що визначаються кінцевими користувачами.

Така зміна позиції розробників викликала в інформаційних системах потужний напрям в теорії, методах і технології розробки систем – моделювання і управління бізнес-процесами, на основі якого були створені усі відомі ERP-системи.

Сьогодні центр додатків аналізу зміщується ще критичніше – у бік маніпулювання роботами, побудови засобів генерування потоків робіт і управління ними. Управління бізнес-процесами і управління потоками робіт – не одне і те саме. Бізнес-процес визначає, що має бути зроблено, у тому числі, встановлюючи співвідношення між входом і виходом даних. У складі бізнес-процесу можуть перебувати складові, що не автоматизуються (або неавтоматизовані), такі, що відпрацьовуються вручну, і він може використовувати будь-які види ресурсів. Потік робіт описує, як може бути досягнутий певний результат.

Проте взаємозв'язок між бізнес-процесом і потоком робіт доволі простий [1–3] і ілюструється схемою на рис. 1.

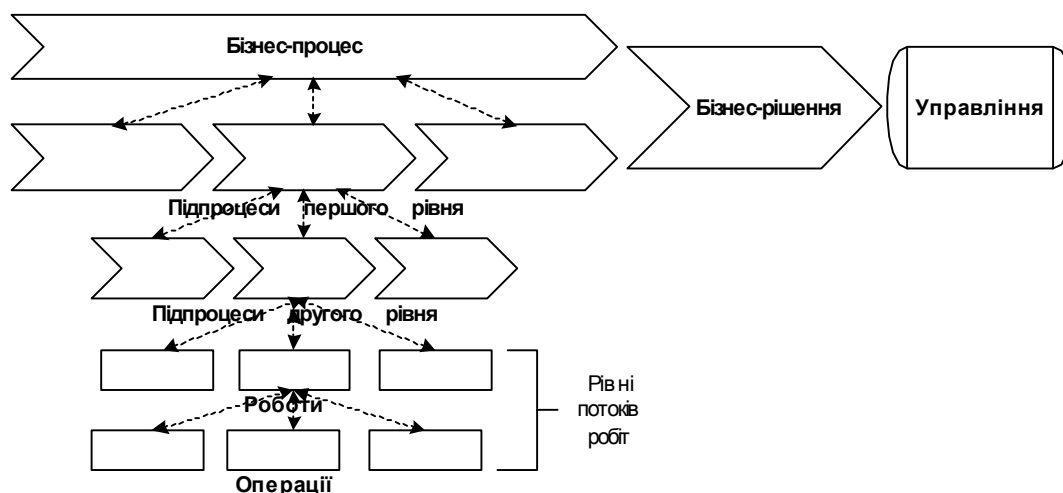


Рис. 1. Взаємозв'язок бізнес-процесів і потоків робіт

Як і в попередньому випадку, зміна позиції розробників пояснюється змінами в концепції побудови інформаційних систем. Перехід, що відбувається у цей час, до орієнтованої для сервісу архітектури (SOA) означає насамперед необхідність виділення функціональних сервісів-елементів SOA, з яких потім здійснюється синтез і модернізація бізнес-процесів.

#### Аналіз останніх досліджень та публікацій

За визначенням, що дається організацією OASIS (Organization for the Advancement of Structured Information Standards), яка займається питаннями стандартизації компонентів SOA, сервісно-орієнтована архітектура є парадигмою розподілених організаційних і утилітарних можливостей, що працюють під управлінням доменів, що належать різним власникам [4].

За визначенням IBM [5], SOA – архітектурний стиль для створення інформаційної архітектури підприємства, що ґрунтується на сервісній орієнтації для досягнення тіснішого взаємозв'язку між бізнесом і підтримувальними інформаційними для бізнесу системами. SOA вводить сервісну орієнтацію як підхід до інтеграції бізнесу на основі пов'язаних між собою сервісів.

У межах концепції SOA крупні конгломерати програмних та інформаційних засобів розділені на невеликі уніфіковані стандартизовані сервіси, кожному з яких, своєю чергою, відповідає базова активність (робота). Для виконання цієї функції використовуються механізми моделювання бізнес-процесів.

У поєднанні з можливостями швидкого внесення змін в процеси наявність таких механізмів вважається основою ефективності інформаційних систем, що ґрунтуються на SOA. Для виконання завдань, що виникають при цьому, використовуються бізнес-інструменти, які загалом забезпечуватимуть:

- побудову нових бізнес-функцій;
- встановлення зв'язків між бізнес-функціями, що виділяються з існуючих застосувань;
- генерування потоків робіт для виконання бізнес-процесів.

У найзагальнішому вигляді SOA передбачає наявність трьох основних учасників: постачальника сервісу – туристичного агента, споживача сервісу – туриста, реєстру сервісів (рис. 2). Взаємодія учасників виглядає доволі просто: постачальник сервісу реєструє свої сервіси у реєстрі, а споживач звертається до реєстру із запитом. Відсутність будь-якого з цих елементів неприпустима, а додавання інших складових на практиці не лише можливе, але й необхідне. Серед таких елементів можуть бути усілякі програмні засоби проміжного застосування, контролюючий порядок і контекст взаємодії, здійснюючий моніторинг і управління сервісами, а також управління метаданими та інші допоміжні процеси.

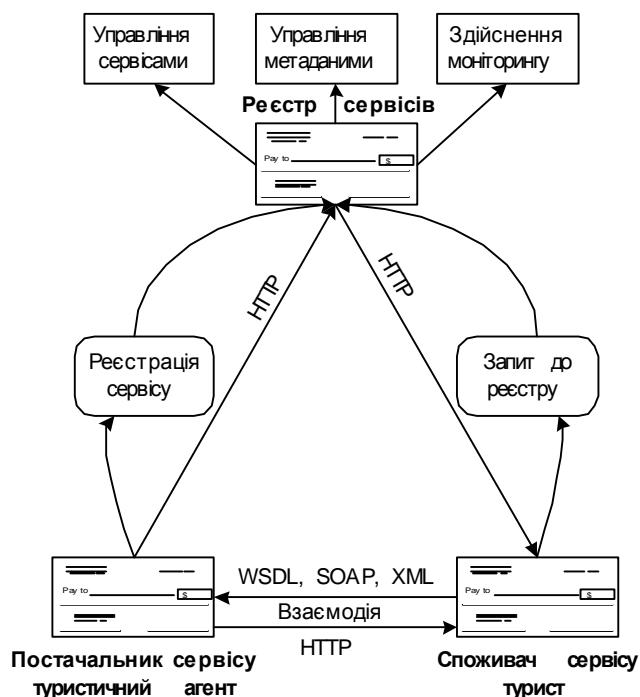


Рис. 2. Загальна схема SOA в туризмі

SOA – це така архітектура додатка, в якій компоненти або сервіси, маючи погоджені загальні інтерфейси, використовують єдині правила (контракт) для визначення того, як викликати сервіси і як вони взаємодіятимуть один з одним [6]. У цій системі поняття “клієнт” і “сервер” абсолютно ситуативні. В одному випадку додаток може працювати, як клієнтський, і викликати зовнішній сервіс, а деякий час після цього, він може стати постачальником сервісів під час звернення до нього іншим застосуванням з метою виконання будь-яких завдань. Якщо SOA розроблена правильно, вона здатна здійснити побудову і підтримку інтеграції, побудованої по архітектурі “клієнт-споживач”. Використовуючи SOA, туристичні організації можуть гнучко і динамічно генерувати нові сервіси, комбінуючи логіку існуючих застосувань і представляючи її за допомогою повторного використання сервісів.

Для інформаційних систем туристичних агентств та фірм виділимо цілі SOA:

- скорочення витрат під час розроблення додатків за рахунок впорядкування процесу розроблення туристичних послуг;
- розширення повторного використання коду;
- незалежність від використовуваних платформ, інструментів, програмних мов розроблення;
- підвищення масштабованості створюваних систем у зв'язку з розвитком туристичного сервісу;

- поліпшення керованості створюваних інформаційних систем.

Принципи SOA:

- архітектура, як така, не прив'язана до якоїсь конкретної технології;
- незалежність організації системи від використовуваної обчислювальної платформи;
- незалежність організації системи від вживаних мов програмування;
- використання сервісів, незалежних від конкретних застосувань, з одноманітними інтерфейсами доступу до них;
- організація сервісів як слабозв'язаних компонентів для побудови систем різнонаправлених по спеціальності туризму організацій.

Сервіс-орієнтована архітектура ґрунтується на таких базових принципах, що уможливають зняти найгостріші проблеми інтеграції додатків [7, 8].

*Слабка зв'язаність.* З точки зору реалізації сервіси в SOA незалежні один від одного: вони виконують певні дії із запитів, отриманих від інших сервісів, і повертають результати. Усі деталі цього виконання повністю приховані: у концепції SOA сервіси – це “чорні ящики”. Тому зміни в реалізації сервісу в жоден спосіб не вплинуть на прикладний компонент, який цей сервіс використовує, і навпаки.

Слабка зв'язаність забезпечує просту і швидко адаптацію системи до змін в структурі і принципах реалізації сервісів.

*“Грубозерниста” структура сервісів.* Сервіси в SOA є модулями бізнес-логіки високого рівня, завдяки чому взаємодія між ними зводиться до обмеженої кількості повідомлень за змістом бізнесу-логіки замість безлічі низькорівневих викликів, що враховують деталі реалізації сервісів. Такий підхід знижує навантаження на мережу і сприяє вищій продуктивності системи. На практиці перехід до SOA може починатися із подання у вигляді сервісів бізнес- або системних функцій низького рівня, які потім завдяки ще одній властивості сервісів – модульності компонуватимуться у високорівневі сервіси, що реалізують певні етапи бізнес- або обчислювального процесу або увесь процес повністю. Важливо ще й те, що з точки зору архітектури сервіс, незалежно від внутрішньої структури і мови реалізації, виглядає як єдине ціле.

Реалізація слабкої зв'язності може ґрунтуватися на поєднанні різних методів і засобів.

*Контракти.* Поняття контракту в SOA трактується дещо ширше, ніж зазвичай. Контракт визначає умови надання і застосування сервісів, включаючи функціональні, технічні (не лише протокол, але й, наприклад, спосіб авторизації) та інші (вартість путівки, гарантії, страхування туриста) параметри і слугує основою угоди між споживачем і провайдером сервісу на усіх етапах його процесу взаємодії. Не дарма одне з поширених визначень провайдера і споживача сервісів ґрунтується на тому, хто з них контракт надає (провайдер-туристичний агент), а хто приймає (споживач-турист).

Контракт дає змогу розділити компоненти на дві частини – публічну, стабільнішу і приховану, таку, яка постійно змінюється. Оскільки спосіб організації компонентів в SOA відповідає способу організації бізнесу або обчислювального процесу, контракт окреслює можливості не стільки програмного компонента, скільки бізнесу або обчислювальної функціональності, розділяючи сервіс на бізнес (обчислювальний) інтерфейс і бізнес (обчислювальний) процес.

Контракти не лише ослаблюють зв'язки між елементами архітектури, але, що не менш важливо, роблять їх явними, упредметненими об'єктами управління.

*Композиція.* Основною перевагою композиції вважається забезпечення можливості багатократного використання раніше реалізованої функціональності. Проте композиція сприяє і адаптивності, оскільки дає змогу швидше проводити багато змін шляхом зміни зв'язків між сервісами. Такі зміни здійснювати набагато легше, ніж змінювати увесь сервіс загалом.

*Динамічне зв'язування.* Спираючись на стабільність контракту, споживач може знаходити необхідні екземпляри сервісів, не перериваючи роботи. Це означає, що система стійка до змін не лише в адресації доступних сервісів (переїзд на інший сервер), але і в їх складі, який може змінитися за час експлуатації. Наприклад, можуть з'явитися нові партнери – туристичні агенти або

відповідальність за надання необхідної бізнес-функції може перейти до іншого провайдера, зокрема зовнішнього, проте це не вплине на роботу споживача. Динамічне зв'язування підтримується реєстром, який виконує функції облікової системи SOA. Переважно він містить посилання на контракти, конкретні екземпляри сервісів, інформацію про їх провайдерів і інколи також про їх споживачів. Все це разом, окрім інших, не менш важливих, функцій, слугує також і конфігурацією SOA, що визначає параметри часу виконання використовуваних сервісів для споживачів сервісів (внаслідок композиції тих, що частенько є також їх провайдерами).

SOA – це набір взаємно сумісних принципів проектування, і динамічне зв'язування часто зараховують до найспецифічніших з них [9]. Переваги динамічного зв'язування особливо необхідні під час роботи великих або децентралізованих інформаційних служб, або коли у взаємодію включається багато сторін, що працюють за ідентичним або за дуже схожим контрактом. Це можуть бути власні філії туристичної компанії (наприклад, роздрібні відділення – ресторани, готелі, санаторії тощо), але найчастіше – це партнери – постачальники туристичних послуг, туристичні агенти тощо.

*Масштабованість інтерфейсів.* За усієї типізації інтерфейсів, правильною практикою вважається залишати можливість для додавання заздалегідь непередбачених елементів даних у передаванні повідомлення. В такий спосіб контракт заздалегідь допускає деяку гнучкість, якщо вона можлива і необхідна в конкретній ситуації.

*Перебудова.* Масштабованість – якість хороша, але недостатня. Інтерфейс завжди повинен якнайповніше відображати, що існують функціональності, крім того, масштабованість не вирішує основних проблем управління змінами. Тому рано чи пізно інтерфейс доводиться міняти, адаптувати до змінених бізнес-вимог. Які процедури при цьому запускаються, кожна компанія (в особі відповідального інформаційного менеджера) сама визначає для своєї організації, залежно від безлічі чинників, не останній з яких – баланс логічності та інтуїтивності в стилі ухвалення рішень. Для стабільніших організацій характерна більша регламентація процесу змін, для гнучкіших – менша.

### **Формування цілей роботи**

**Мета роботи** – підвищити гнучкість інформаційних систем у туристичній сфері з даними, знизити витрати на розроблення додатків, збільшити швидкість реагування на постійні зміни вимог туристичного бізнесу, а також забезпечити необхідний рівень інтеграції між інформаційними системами за допомогою SOA – сервісно-орієнтованої архітектури.

### **Основний матеріал**

#### **1. Стандарти Web-сервісів, які можна використати у туристичній сфері**

Для успішної інтеграції систем на основі Web-сервісів використовується багато стандартів, основними з яких є:

- BPEL – мова реалізації бізнес-процесів;
- WSDL – мова опису сервісів;
- SOAP – протокол обміну повідомленнями;
- UDDI – універсальний формат каталогу для пошуку та інтеграції Web-сервісів.

Композиція Web-сервісів на основі елементарних робіт здійснюється за допомогою мови реалізації бізнес-процесів – Business Process Execution Language (BPEL). Вона з'явилася як результат об'єднання мови WSFL (Web Services Flow Language), розробленої корпорацією IBM, і мови XLANG, створеної в Microsoft. Мова розроблена на нотації XML.

Використання цієї мови дає змогу здійснити формування і виконання потоку робіт як послідовність логічних дій, що включають:

- прийняття запиту на включення роботи в процес;
- перевірка опису, і в разі збігу параметрів – підготовку позитивного відгуку на запит;
- відхилення запиту з видачею обґрунтування.

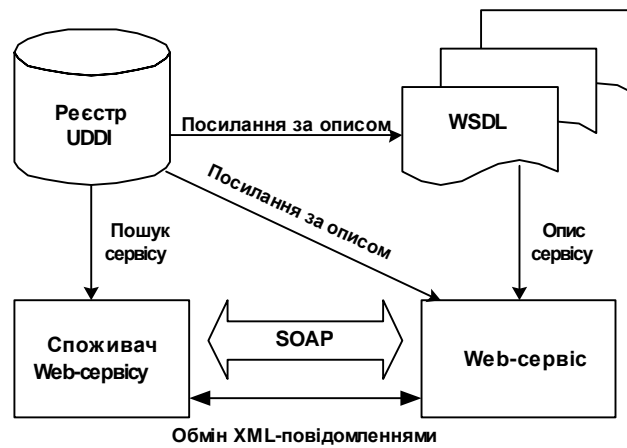


Рис. 3. Стандартизація типів сервісів

Як правило, BPEL формує потік робіт, що складається з послідовностей логічних дій або активних операцій, кожній з яких відповідає свій квадратик на діаграмі потоку робіт або функція програмного коду. При цьому можливі дві форми використання BPEL:

1. Виконуваний BPEL – процес, який також розглядається як сервіс, і може бути вузлом оркестрування. Програмні продукти, що реалізують виконуваний BPEL-процес, називаються BPEL-engine (двигун BPEL-процесу). Тут один виконуваний процес може включати інший, що дає ефект включення одного оркестрування (послідовності сервісів) в інший, як це показано на рис. 4.

2. Абстрактний процес є майже повністю ідентичний до виконуваного процесу, за винятком наповнення даними. У цій якості він являє собою логіку бізнес-процесу і використовується з такою метою:

- визначає поведінку елементів організаційної структури, що підтримує процес;
- є керівництвом для програмістів і розробників, що автоматизують процес;
- вважається входом для комерційного програмного забезпечення, що утворює структуру системи.

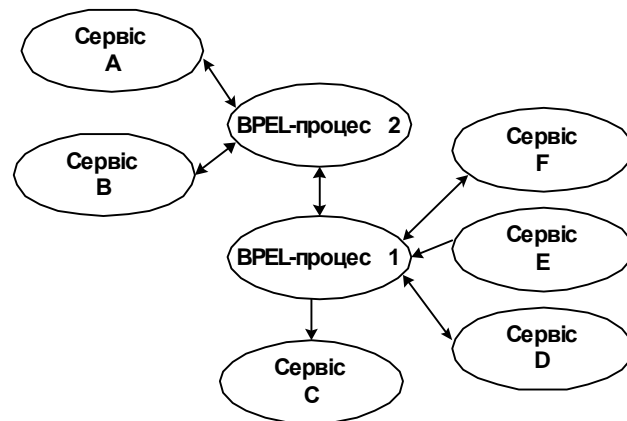


Рис. 4. Схема зв'язаного оркестрування

WSDL (Web Services Description Language) – мова опису Web-сервера-сервісів, що ґрунтується на мові XML. Забезпечує правильний вибір сервісів для передачі від провайдера до споживача. Інформація у форматі WSDL використовується розробниками сервісів для включення сервісів у систему.

SOAP (Simple Object Access Protocol) – протокол обміну повідомленнями, також написаний у форматі XML, призначений для передачі даних із Web-сервісів. Файли SOAP, що створюються автоматично, включають дані з опису сервісів у форматі WSDL.

UDDI є набором правил реєстрації і витягання даних про наявні сервіси. Розробляючи програми, програмісти можуть здійснювати пошук в реєстрі UDDI необхідних сервісів для

включення їх у програми. Цей реєстр може бути необхідний в процесі виконання програми, якій необхідні сервіси, що надають дані, наприклад, про вартість туристичних послуг або путівки.

Розглянемо здійснення уніфікації подання бізнес-процесів, які були зроблені компанією IBM, що впровадила у практику автоматизації уніфікований опис процесу – Rational Unified Process (RUP). Приклад створення автоматизованої системи з використанням RUP і програмних продуктів IBM, що підтримують моделювання потоку робіт в туризмі, показаний на рис. 5.

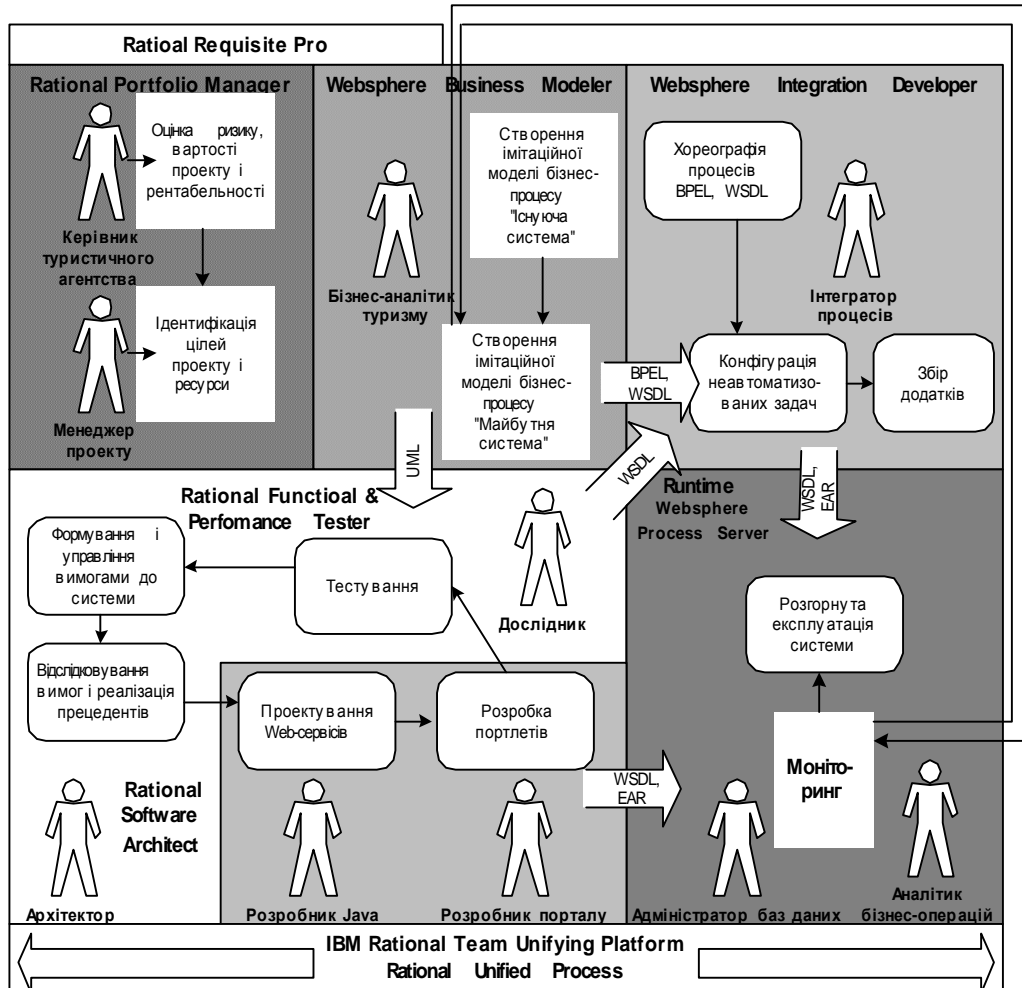


Рис. 5. Створення автоматизованої системи з використанням уніфікованого процесу Rational Unified Process для туристичної сфери

Нижче наведено покроковий опис показаного на цьому рисунку уніфікованого процесу створення інформаційної системи.

Крок 1. Додаток Rational Portfolio Manager допомагає досягти узгодження на верхньому рівні моделювання за такими показниками, як рентабельність (ROI), вартість та інших умовах ведення бізнесу.

Крок 2. Бізнес-аналітик розробляє модель бізнес-процесу, використовуючи додаток Websphere Business Modeler.

Крок 3. Бізнес-процес подається в Rational Software Architect у стандартній нотації UML.

Крок 4. Rational Software Architect перетворює модель в нотації UML у код Java.

Крок 5. IBM Rational Application Developer автоматизує процес створення нових Web-сервісів на основі існуючих ресурсів, таких як компоненти Javabeans або EJB, за допомогою автоматичного генерування файлів WSDL, що описують Web-сервіси, дескриптора SOAP і клієнта для тестування Web-сервісів.

Наприклад, у формі портлета. Портлет – стандартний портальний компонент призначений для користувача інтерфейсу, що надбудовується (Wikipedia). Портлет виділяє фрагменти розмітки, які вбудовуються в сторінку порталу. Можна уявити собі сторінку порталу як колекцію портлетних вікон, що не перекриваються, кожне з яких відображає портлет. Портлет є ніби веб-сервер-додатком, розміщеним на порталі. Приклади портлетів: e-mail, повідомлення про авіа-рейси, останні туристичні новини та акції.

Завдяки існуючим стандартам портлетів розробники можуть створювати портлети, що вбудовуються у будь-який портал, що відповідає стандартам.

Крок 6. Websphere Integration Developer реалізує бізнес-процес у форматі BPEL. Розробник, що здійснює інтеграцію web-сервісів в бізнес-процес, може витягувати готовий сервіс з UDDI і включити його в бізнес-процес, або включити в нього функції, що виконуються людьми. Отримуваний результат розгортається у вигляді додатка Websphere Process Server.

## 2. Програмні засоби реалізації Web-сервісів

Під час реалізації проектів, в яких використовується управління процесом виконання, керівник проекту і члени команди для підтримки розробки застосовують Rational Clearquest. Цей продукт автоматизує призначення завдань і відстежує хід їхнього виконання, а також використовується для тестування і виявлення дефектів у спроектованому потоці робіт. Він також застосовується під час редагування, сертифікації і верифікації сервісів.

Для виконання завдань управління конфігурацією застосовується IBM Rational Clearcase, інтегрований з Eclipse.

Часто аргументом проти переходу до SOA є твердження, що він пов'язаний з повним зломом усієї існуючої системи, і тому вимагає одноразових, дуже високих витрат. Насправді перехід до SOA не вимагає одноразової і повної зміни існуючої інформаційної системи. Окремі кроки або додатки можуть впроваджуватися поступово, і увесь перехід може бути здійснений еволюційним шляхом. При цьому використовуються технології, що забезпечують дотримання раніше введених стандартів, а також спадковість по відношенню до раніше інсталюваних працездатних програмних продуктів і технічного устаткування. Узагальнену еволюційну структуру SOA в туризмі показано на рис. 5.

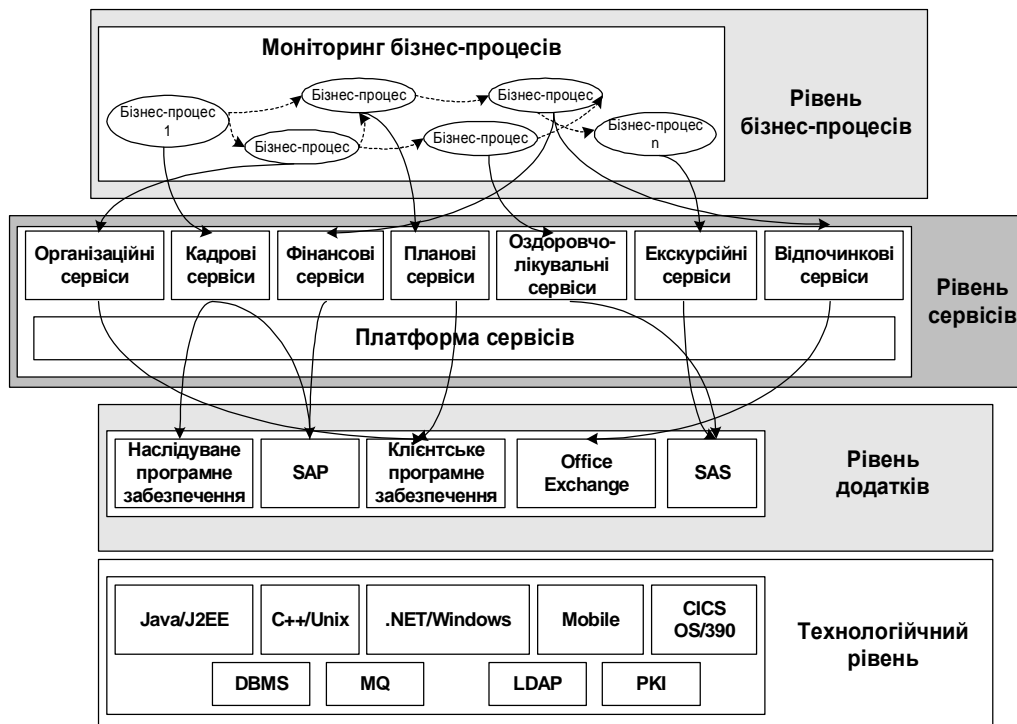


Рис. 6. Приклад еволюційної сервіс-орієнтованої архітектури в туризмі



На рис. 6 показано типову ситуацію, коли інформаційна система на базі SOA будується на основі існуючого пропріативного програмного забезпечення. Як приклад, показані такі популярні застосування, як SAP, SAS, офісні пакети, а також клієнтське програмне забезпечення і програми, успадковані від попередніх етапів розвитку системи.

Уся структура складається з чотирьох рівнів: бізнес-процесів, сервісів, додатків і технологічного рівня. Багато аналітиків виводять історію SOA, починаючи з кінця минулого століття, маючи на увазі розвиток функціонального підходу, а пізніше концепцію COBRA і інструментарій Java. Проте поширення вона не набула, оскільки у складі використовуваної у той час архітектури був відсутній сервісний рівень. Множинні застосування мали різний інтерфейс GUI, несумісну логіку бізнес-процесів, різні системи зберігання даних (СУБД). Усе це утруднювало роботу системи. Реалізація бізнес-процесів безпосередньо була пов'язана з технологічним рівнем і залежала від швидкості обміну інформацією між однаковими за функціональним призначенням, але різномірними за конструкцією компонентами (наприклад, між різними БД додатків).

Перехід до сучасної SOA і включення в неї додаткового сервісного рівня формує лінію автономних сервісів, кожен з яких прив'язаний до відповідного бізнес-домена і платформи Web-сервісів, що дає змогу здійснювати їх використання значною мірою незалежно від розташованих нижче за рівнем додатків і технологічної платформи.

Введення рівня сервісів створює також зручні умови для побудови розташованого вище від рівня бізнес-процесів і їх функціонування, а саме:

- лінія бізнес-сервісів відтворює грубий набір функцій, які є готові для включення в бізнес-процес;
- опис (contract) кожного сервісу в лінії сервісів однозначно його визначає, формуючи інтерфейс. Наявність останнього дає можливість створювати бізнес-процеси незалежно і без залучення знань стосовно технологічної платформи;
- інструмент реєстру і пошуку сервісів підтримує динаміку і можливість змін бізнес-процесів шляхом оперативного доступу до сервісів у разі такої необхідності;
- модель даних, реалізована на рівні сервісів, спирається на структуру бізнес-домена і теж не залежить від моделей даних окремих застосувань. Оскільки XML застосовується як канонічний формат обміну даними, включення сервісу в бізнес-процес здійснюється також незалежно від внутрішньої структури даних у додатках;
- модель захисту сервісів, що діє в межах цього рівня, забезпечує універсальний контроль рольового розподілу і підключення сервісу, авторизуючи використання сервісу деяким процесом. Це також уможливує уникнути труднощів безпосередньої взаємодії із засобами захисту додатків, з їх відмінністю в принципах організації, інтерфейсах тощо;
- модель управління рівнем обслуговування генерує статистику використання сервісів, що є частиною моніторингу бізнес-процесу, що виконується верхнім рівнем (бізнес-процесів).

Відсутність цього рівня у попередніх поколіннях інформаційних систем створювала безліч проблем, які сприяли швидкій деградації бізнес-процесів, і саме цей чинник гальмував поширення SOA [9].

Наведемо приклад, що відтворює можливість переходу до SOA в туристичній сфері шляхом еволюції і удосконалення системи, що не вимагає разових великих витрат і повного знищення звичного процесу функціонування системи. У цьому прикладі перехід до SOA здійснюється поетапно (рис. 7).

На рис. 7 в системі, що розвивається, наведемо приклад двох підсистем – кадрів і фінансів та проаналізуємо їхні бази даних. На першому етапі еволюції розробляються нові сервіси, що включаються у ці підсистеми. Нові сервіси відразу ж виконані під стандарти і вимоги майбутньої SOA. Далі від них показані сервіси, побудовані на основі успадкованих застосувань. Ці сервіси зберігають конструктивні особливості старої системи. Для їхнього використання у майбутній системі вони об'єднуються в спільну оболонку, яка подається як інтерфейс, спроектований вже під нову логіку реалізації процесів. Потім слідує так звані композитні сервіси, що будуються як комбінація атомарних сервісів, в яких можуть бути задіяні як нові, так і успадковані сервіси.

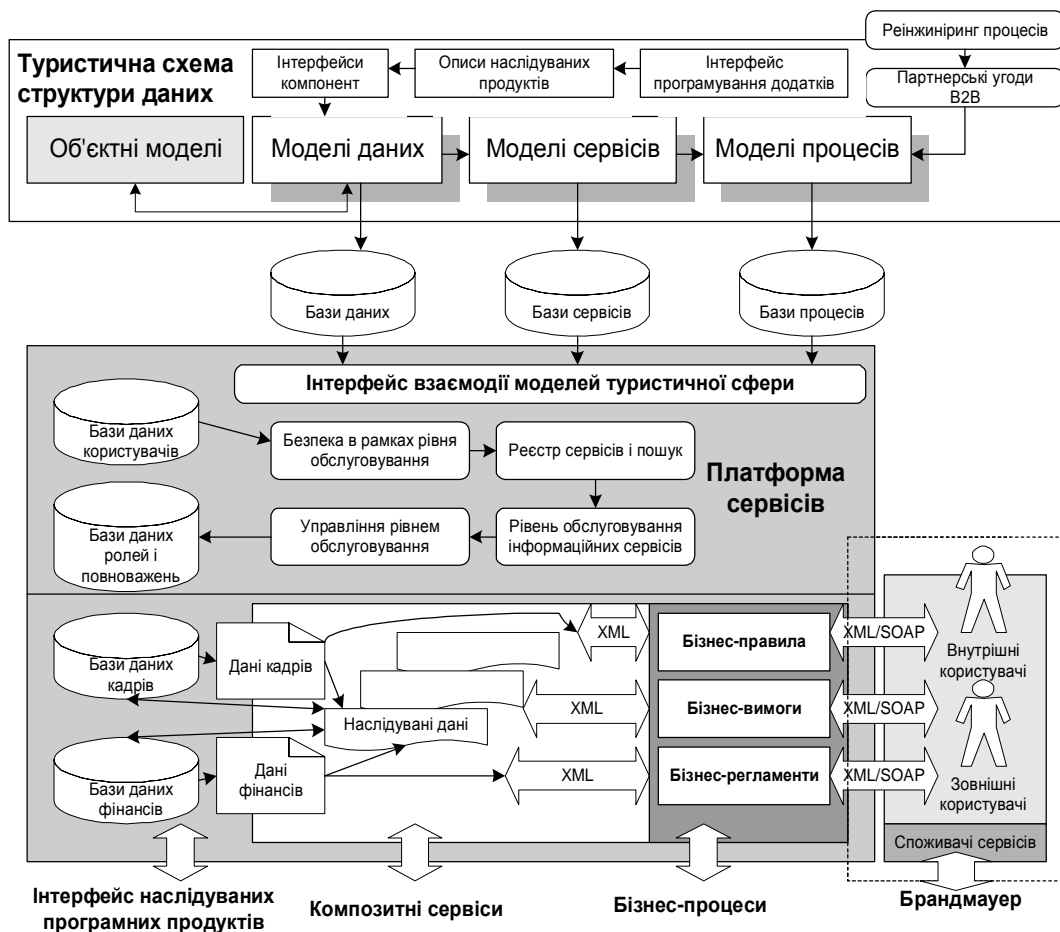


Рис. 7. Еволюція інформаційної системи на основі SOA в туристичній сфері з включенням успадкованого програмного забезпечення

За допомогою інструментарію, включеного в рівень сервісів (реєстру, служб управління рівнем обслуговування, безпеки і побудови бізнес-процесів), будується бізнес-процес, що об'єднує як нові сервіси, так і успадковане програмне забезпечення.

Розробка подібної схеми може слугувати основою організації і планування процесу переходу до SOA в туристичній сфері за допомогою поступового розвитку інформаційної системи.

### Висновки

У роботі описана методика сервісно-орієнтованої архітектури для створення інформаційної архітектури туристичної сфери, що ґрунтується на сервісній орієнтації для досягнення тіснішого взаємозв'язку між бізнесом і підтримувальними інформаційними для бізнесу системами. Запропоновано застосування сервісної орієнтації SOA як підхід до інтеграції туристичного бізнесу на основі зв'язаних між собою сервісів шляхом еволюції і удосконалення системи, що не вимагає разових великих витрат і повного знищення звичного процесу функціонування системи.

1. Кузнецов С. Р. SOA: просто для большинства, сложно для меньшинства // IEEE Computer Society. – 2007. – № 11.
2. Mathias Weske. Business Process Management. Springer-Verlag, Berlin-Heidelberg, (2008).
3. Ричард Вурт Конвергенция виртуализации, grid и SOA // Открытые системы. – 2008. – № 3.
4. Bobby Woolf Exploring IBM SOA. Technology&Practice. Maximum press (2009).
5. IBM Redbook. Patterns: Implementing an SOA Using an Enterprise Service Bus [Електронний ресурс]. – Режим доступу: <http://www.ibm.com/Index.cfm? ArticleID=22084>.
6. Плющенко П.А. Способы решения интеграционных задач// Директор информационной службы. – 2006. – № 4.
7. IBM Redbook Patterns: Service-Oriented Architecture and Web Services. [Електронний ресурс]. – Режим доступу: <http://www.redbooks.ibm.com/abstracts/sg246303.html>.
8. Плющенко П.А. Автоматизация

управлення бизнес-процессами// Промышленные АСУ и контроллеры. – 2006. – № 12. 9. SOA Development Using the IBM Rational Software Development Platform: A Practical Guide. [Електронний ресурс]. – Режим доступу: [http://www07.ibm.com/sg/soa/downloads/SOA\\_Development\\_using\\_Rational\\_Software](http://www07.ibm.com/sg/soa/downloads/SOA_Development_using_Rational_Software).

УДК 004.78

Л.Б. Чирун, О.Р. Гарасим

Національний університет “Львівська політехніка”,  
кафедра інформаційних систем та мереж

## АНАЛІЗ ЗАХИЩЕНОСТІ МЕРЕЖІ НА ОСНОВІ КОНСОЛІДОВАНИХ РЕСУРСІВ ЗАГРОЗ

© Чирун Л. Б., Гарасим О.Р., 2011

**Проведено аналіз захищеності мережі з використанням консолідованих ресурсів загроз, які були виявлені й зареєстровані в БД. Складено діаграму аналізу загроз та оцінено ризики поширення їх у мережі, запропоновано виявлення слабких ланок мережі.**

**Ключові слова:** загроза, мережа, ризик, консолідація.

**In the paper conducted analysis the network security using a consolidated threats resources that were found and registered in the database. Build diagram of analysis threats and assessed risks in distribution network, suggested identifying weak links in the network.**

**Key words:** threat, network, risk, consolidation.

### Вступ. Загальна постановка проблеми

Діяльність будь-якої корпорації тісно пов'язана з використанням інформаційних мереж зв'язку, які будуються із застосуванням електронних технологій передавання, збереження, опрацювання, використання корпоративної інформації. Надійне функціонування цих систем безпосередньо впливає на економічну діяльність та фінансовий стан корпорації. В управлінні корпоративною діяльністю разом із фінансовими ризиками необхідно враховувати і ті, які пов'язані із використанням інформаційних систем. Тому для управління ризиками повинна проводитися консолідація інформації системи обліку і вивчення усіх подій, що спричиняють збитки, визначення ймовірностей їх настання, ризики поширення, способи їх упередження. Це завдання є надзвичайно важливим у сучасній корпоративній діяльності, його виконання має першочергове значення.

Система менеджменту інформаційної безпеки корпоративної мережі пов'язана із впливом різних чинників діяльності користувачів мережі і є основою економічної стабільності та збереження високого рівня безпеки корпорації. Для захисту корпоративної інформації, особливо конфіденційної, адміністраторам необхідно приймати своєчасні та зважені управлінські рішення, опрацьовуючи консолідовану інформацію загроз та слабких місць мережі.

Консолідована інформація діяльності системи безпеки корпоративної мережі дає змогу отримати вичерпну інформацію про стан мережі та здійснювати ефективний моніторинг подій, виявляти атаки, несправності та слабкі місця, ізолювати загрози безпеці корпоративної інформації. На основі консолідованої інформації проводиться діагностика, контроль та адаптація менеджменту інформаційної безпеки, проведення прямого контролю безпеки. Адаптація менеджменту інформаційної безпеки необхідна для задоволення бажаних результатів, незважаючи на зміну цілей управління корпорації, технологічних умов або розширення діяльності корпорації.