

11. П'ятницька-Позднякова І. С. Основи наукових досліджень у вищій школі : навч. посіб. / І. С. П'ятницька-Позднякова. - К., 2003. - 116 с.

12. Посібник з питань партисипативної демократії (демократії участі) на місцевому рівні / В. В. Толкованов, А. К. Гук, В. М. Олуйко, В. Я. Прошко / за ред. В. В. Толкованова. - К. : Крамар, 2011. - 199 с.

13. Фукуяма Фр. Доверие: социальные добродетели и путь к процветанию / Фр. Фукуяма ; пер. с англ. - М. : ООО "Изд-во АСТ" : ЗАО НПП "Ермак", 2004. - 730 с.

**Олександр Шевцов,**  
здобувач кафедри інформаційної політики  
та технологій НАДУ,  
начальник відділу розробки  
та впровадження системи  
"Електронного кабінету платника податків"  
Департаменту розвитку та модернізації  
Державної податкової служби України

### **Технологічні аспекти впровадження концепції сервіс-орієнтованої архітектури у сфері державних електронних послуг**

У статті проаналізовано загальні технологічні аспекти впровадження концепції сервіс-орієнтованої архітектури (SOA) у сфері державних електронних послуг, що дає змогу на сьогодні розв'язати одну з найбільш нагальних та складних проблем в ІТ-сфері - проблему реінжинірингу бізнес-процесів, інтеграції програмних компонент та інформаційних застосувань при впровадженні складних, інтегрованих ІТ-архітектур в організації. Розглянуто основні компоненти SOA, наведено їх загальну характеристику та попередньо проведено порівняння концепції SOA з іншими технологічними засобами розробки та інтеграції ІС.  
*Ключові слова:* сервіси, інформаційна система, бізнес-процеси, інтегрована архітектура електронних комунікацій, компоненти інтеграції.

The general technological aspects of service-oriented architecture (SOA) in the state electronic services are analyzed in the article. It helps to solve one of the most important and complex problems in IT-services - the issue of business-processes re-engineering, the integration of software component- and information usage while introducing complex and integrated IT-architectures in the organization. The basic SOA components are reviewed in the article, their main characteristics are given. SOA concepts are compared with other technological ways of IS development and integration.

*Key words:* services, information system, business processes, integrated architecture of e-communications, integration components.

*Постановка проблеми.* Протягом всієї історії розробки та впровадження інформаційних систем (далі - ІС) однією з головних була проблема інтеграції та повторного використання програмних компонент. На початку ідея повторного використання була реалізована за рахунок вбудованих процедур та функцій, які потім оформлялися в програмні модулі. Наступним етапом реалізації методу повторного використання стала компонентна архітектура, яка розглядає компонент як модуль, що реалізує стандартний інтерфейс. Такий підхід значно прискорює процеси розробки та функціонування застосувань, у тому числі їх впровадження. До такої компонентної архітектури належать платформи NET [1] та Java 2 Enterprise Edition (J2EE) [2]. Хоча компонентна архітектура забезпечує швидку та зручну інтеграцію компонентів, однак такий підхід стосується тільки програмних модулів, які були розроблені відповідно до специфікацій зазначених платформ.

Наступним етапом удосконалення процесів стала концепція "бізнес на вимогу". Відповідно до такої концепції всі внутрішні бізнес-процеси тісно інтегровані всередині організації, а також із бізнес-процесами сторонніх споживачів, що дає можливість гнучко та швидко реагувати на кон'юнктуру ринку та зовнішні впливи. Однак існували й загальні обмеження реалізації такого підходу, а саме: використання єдиної платформи є занадто дорогим рішенням, при розробці нового або модифікації існуючого ПЗ доцільно максимумно використовувати вже наявні компоненти тощо.

Необхідність вирішення таких обмежень призвела до появи концепції сервісно-орієнтованої архітектури (Service-Oriented Architecture, SOA) [3]. Поява SOA стала закономірним кроком на шляху пошуку розв'язання однієї з найбільш нагальних та складних проблем у ІТ-галузі - проблем реінжинірингу бізнес-процесів, інтеграції програмних компонент та ІС при побудові складних ІТ-архітектур. SOA-модульний підхід до розробки інтегрованої архітектури електронних комунікацій, який являє собою модель взаємодії компонентів ІС, заснований на використанні сервісів зі стандартизованими інтерфейсами.

*Аналіз останніх досліджень та публікацій.* Публікації, присвячені розв'язанню проблеми практичного використання технологічних аспектів концепції SOA при реалізації інтегрованої архітектури на підприємстві, у вітчизняній науковій літературі не зустрічаються. Реальні проекти з впровадження концепції SOA в організаціях на сьогодні в Україні майже відсутні.

Деякі аспекти переходу до використання концепції SOA при впровадженні цільової ІТ-архітектури органів Державної податкової служби розглядалися під час проведення науково-практичних семінарів у Національному університеті Державної податкової служби України [4; 5].

Останніми роками провідні зарубіжні ІТ-компанії, як-от IBM, Microsoft, Oracle, Sun, RedHat приділяють значну увагу проблемам розвитку та ефек-

тивного використання концепції SOA при впровадженні інтегрованої ІТ-архітектури в організації (у тому числі і в органах державної влади).

*Невирішені раніше частини загальної проблеми.* Незважаючи на активні дослідження проблем інтеграції ІС та програмних модулів, у працях науковців недостатньо уваги приділяється питанню вивчення та визначення практичних підходів до побудови інтегрованої архітектури електронних комунікацій із використанням концепції побудови SOA. Тому вивчення й визначення переліку головних технічних аспектів реалізації сервіс-орієнтованого підходу при впровадженні інтегрованої архітектури в структурі електронних інформаційних комунікацій органів державної влади потребує додаткового аналізу та наукових досліджень.

*Метою статті* є попередній аналіз технологічних аспектів впровадження концепції сервіс-орієнтованої архітектури у сфері надання державних електронних послуг при реалізації інтегрованої архітектури електронних комунікацій в організації.

*Виклад основного матеріалу.* В концепції SOA використовуються чотири основні технологічні складові, а саме: сервіси, репозиторій описів сервісів, "фронт-енд" застосування та сервісна шина.

### **1. Сервіси - головний компонент SOA**

Сервіси відрізняються від звичайних об'єктних компонентів (наприклад Common Object Request Broker Architecture (CORBA) та Enterprise JavaBeans (EJB)) тим, що вони не тільки орієнтовані на обслуговування запитів, а також включають великі та логічно ізольовані блоки бізнес-функціоналу. На відміну від звичайних об'єктних рішень, які складаються з низки невеликих елементів, SOA будуються з малої кількості великих елементів. Таке "збільшення" істотно знижує навантаження на розробника з контролю над взаємозалежністю компонентів та сприяє їх повторному використанню. Усі взаємозв'язки "невеликих" компонентів відбуваються всередині "великих" блоків. Домогтися такої ізольованості складно, особливо на ранніх етапах впровадження концепції SOA. Для досягнення цієї мети необхідним є проведення декомпозиції готових застосувань та використовуваних ними баз даних. У цьому випадку переходять до побудови "фасадів" - незалежних сервісів, що включають роздрібненість та надмірність інформації в різних базах даних. При цьому "фасади" стають єдиним дозволеним способом доступу ззовні до цих даних на рівні концепції SOA.

### **2. Репозиторій описів сервісів**

Опис сервісу може бути представлено у форматі WSDL (Web Service Definition Language). Усі описи сервісів (опис інтерфейсу) необхідно зберігати в репозиторії. Такий репозиторій є каталогом сервісів. SOA не накладає будь-яких обмежень на технологію, що використовується для реалізації конкретного репозиторію, але при цьому є важливими два моменти: описи

всіх інтерфейсів необхідно уніфікувати; репозиторій повинен бути єдиним по всій організації, при цьому копії репозиторію можуть бути розосереджені територіально, а управління частинами репозиторію може бути делеговане різним групам.

Згодом компанії Microsoft та Ariba виступили з ініціативою у створенні проекту UDDI (Universal Description, Discovery and Integration) [6]. Провайдери сервісів публікують свої сервіси в UDDI, де ініціатори запитів можуть знайти ці сервіси та встановити зв'язок з ними за допомогою визначених механізмів. У UDDI виділяються чотири основні структури даних: Business Entity, Business Service, Binding Template та TModel. Елемент Business Entity відповідає організації, яка пропонує сервіс, зазначений в елементі Business Service. Елемент Binding Template визначає спосіб, яким споживач сервісу може зв'язатися із цим сервісом. Елемент TModel являє собою технічну модель сервісу.

### **3. Клієнти сервісів**

Головними клієнтами сервісів є так звані "фронт-енди" - різноманітні клієнтські застосування, що функціонують на Java-серверах застосувань або на серверах IIS/.Net. "Фронт-енд" повинен знати та виконувати контракт, якому задовольняє сервіс, що викликається. Ця функціональність покладається на розробника "фронт-енду". Якщо "фронт-енд" не може виконати контракт сервісу, то сервіс не повинен його обслуговувати. Але до звичайних сервісів можуть звертатися і високорівневі сервіси, наприклад застосування, що оформлене як сервіс ділового процесу.

### **4. Сервісна шина**

Ідея SOA полягає в тому, щоб забезпечити IT-інфраструктуру від зміни поколінь IT та інтегрувати між собою різноманітні успадковані технології. У цьому разі важливим є компонент SOA як шина сервісів. Головними її завданнями є технологічна інтеграція систем, що підключаються, а також надання інтегрованого підходу до журналювання операцій та засобів забезпечення безпеки. В рамках концепції SOA сервісна шина може бути представлена як набір паралельних шин, що відповідають різним принципам комунікації, наприклад асинхронним (Message-Oriented Middleware, MOM), синхронним (CORBA, RPC), FTP-передачам тощо. Такі шини можуть існувати цілком самостійно на різних рівнях ієрархії корпоративної інфраструктури.

### **5. Управління Web-сервісами**

Управління Web-сервісами виникає в той момент, коли в організації, в рамках реалізації SOA переходять до масштабного розгортання Web-сервісів, які інкапсулюють логіку не тільки застосувань внутрішнього користування, але й зовнішніх (клієнтських) систем. Роль системи управління в даному випадку полягає в забезпеченні постійного моніторингу та контролю над

працюючими в SOA застосуваннями. Управління Web-сервісами включає в себе більшість класичних функцій системного управління: моніторинг, генерацію повідомлень для системного адміністратора, управління рівнем обслуговування (Service Level Agreement, SLA) та якістю сервісу, обробку важливих ситуацій, аналіз ключових причин збоїв. Системи управління Web-сервісами засновані на таких базових стандартах, як XML (eXtensible Markup Language), Simple Object Access Protocol (SOAP) [7] та WSDL, які здатні діагностувати не тільки типи, але й семантику запитів. Крім розширених можливостей системного управління в компетенцію управління Web-сервісами входить забезпечення і таких завдань, як управління життєвим циклом Web-сервісів та організація захисту в середовищі SOA.

#### **6. Безпека та підтримка цілісності Web-сервісів**

Для забезпечення конфіденційності та цілісності повідомлення, що передається між сервісами, до механізмів безпеки пред'являються підвищені вимоги. Стандарт SOAP вимагає, щоб конверт, заголовок та тіло повідомлення представлялися в незашифрованому вигляді. Підписи виконують кілька важливих завдань для забезпечення всебічної безпеки в рамках такої архітектури. Цифрові підписи дають змогу перевірити цілісність окремих блоків даних та розпізнати маніпуляції з повідомленнями (зміна, видалення або зміна даних). Для підтвердження автентичності всього повідомлення потрібно забезпечити єдність окремих його блоків. У результаті здійснюється криптографічний зв'язок блоків, що не залежать від їх положення всередині повідомлення. У процесі транспортування повідомлення загальна контрольна сума захищається від зміни за допомогою криптографічного механізму шифрування. Крім перевірки цілісності даних та автентичності повідомлень, підписи надають можливість аутентифікації відправника повідомлення або його частин.

#### **7. Моделювання та управління якістю бізнес-процесів**

Ефективність SOA полягає в здатності забезпечити динамічність бізнесу за допомогою інтеграції та багаторазового використання бізнес-процесів. Сполучною складовою між бізнес-вимогами та ІТ-рішенням, що розгортається, та яке засноване на використанні сервісів, може стати принцип моделювання. Це дає змогу здійснити опис бізнес-вимог, перевести розробку на більш високий рівень, абстрагуватися від деталей реалізації та зосередитися на бізнес-сервісах. У поєднанні з постійним моніторингом та реінжинірингом ці підходи утворюють основу методології BPM (Business Process Management) [8].

Будь-які ІС, що побудовані в SOA, кожна з яких при цьому перебуває в різному підпорядкуванні, можуть бути об'єднані в рамках того чи іншого бізнес-процесу без попереднього виконання додаткових процесів інтеграції. Завдяки SOA принцип модульності отримує нову форму та нові якості, які

набагато збільшують її ефективність, внаслідок розширення сфери застосування в окремих ІС. Як наслідок, інтеграція застосувань усе частіше ґрунтується на підходах SOA та технологіях Web-сервісів, враховуючи переваги надання бізнес-функціональності загальнодоступними сервісами. З цієї причини SOA може бути потужним каталізатором розвитку BPM. Архітектурний стиль SOA заснований на модульності та композиції, через що його можна вважати відповідною основою для такої інфраструктури.

#### **8. Композиція та побудова бізнес-процесів**

SOA та BPM методологічно доповнюють один одного. Розглянемо структуру сервісу. Сервіси облікових систем надають функції збереження - вибірки інформації. Такі сервіси називають "дрібнозернистими" через їх низький рівень абстракції та високу детальність. З них можна формувати "крупнозернисті" сервіси, які, у свою чергу, можуть компонуватися в бізнес-сервіси більш високого рівня. Звичайно, зовсім не обов'язково, щоб сервіси були організовані в окремі прошарки. Важливим є сам принцип узагальнення - від специфічних функцій певних систем до композитних сервісів рівня бізнес-архітектури. Саме наявність слабо зв'язаних сервісів на різних рівнях дає змогу гарантувати гнучкість за рахунок реорганізації зв'язків.

Інструменти BPM представляють бізнес-процеси у вигляді діаграм, що складаються з послідовностей дій, до кожного з яких бізнес-аналітики прив'язують ті чи інші операційні еквіваленти з ІТ. Такими операційними еквівалентами можуть бути і сервіси. Деякі з цих сервісів можуть бути самостійними бізнес-процесами, внаслідок чого бізнес-аналітики можуть спеціалізуватися на окремих предметних областях в організації. Крім власне опису бізнес-процесів, для реалізації BPM необхідно також мати можливість точно представляти ці описи у вигляді ІТ-активів, а також розвивати їх у міру зміни бізнес-процесу. З урахуванням цього була створена мова BPEL (Business Process Execution Language) [9], яка спрямована на формування сервісів шляхом рекурсивної композиції та поєднання моделі процесу проектування з часовою моделлю виконання в єдиному описі.

#### **9. Опис бізнес-процесів**

Модель процесу в BPEL є прямим розширенням сервісної моделі WSDL. Як мова опису інтерфейсів сервісів, WSDL не прив'язаний до якогось протоколу, хоча найчастіше протоколом доступу до сервісу є SOAP. SOAP, у свою чергу, не передбачає наявності WSDL. Більше того, WSDL не дозволяє повністю описати модель передачі повідомлень SOAP. BPEL компенсує це в частині свого розширення WSDL, шляхом надання моделі високорівневого протоколу обміну повідомленнями між сервісами.

З точки зору технології реалізації, теоретично може бути використана будь-яка мова програмування та будь-яка платформа, що підтримує модель програмування BPEL. Практично на сьогодні це сам BPEL з його XML-слов-

ником. Вибір контейнерів, що підтримують "виконуваний" BPEL, дуже широкий та включає як комерційні реалізації (у тому числі від всіх провідних розробників платформ), так і некомерційні з відкритим вихідним кодом.

### **10. Моніторинг та оцінка ефективності роботи процесів**

У SOA процес моніторингу бізнес-активності здійснюється за допомогою спеціалізованих застосувань Business Activity Monitoring (BAM). На сьогодні BAM-функціональність стала стандартним компонентом для управління бізнес-процесами (BPM). Як частина пакета BPM BAM-застосування мають доступ до багатьох систем, які можуть бути залучені до виконання зазначеного бізнес-процесу. BAM-застосування поєднує цю інформацію з даними прогнозуючих систем та сховищ даних, отримуючи складні звіти як по кожному окремому користувачу, так і щодо діяльності організації в цілому. Таким чином, за допомогою налаштованого під конкретні завдання інструментарію BAM організація може отримувати різну інформацію про свою діяльність.

Крім того, BAM-застосування має доступ до виконуваних бізнес-процесів у тому випадку, якщо воно обмежене пов'язаними з ним моделями процесів. BAM отримує негайну інформацію про відхилення в процесах та може надіслати попередження. Отримана таким чином інформація дає змогу проводити оцінку персоналу, враховуючи зону відповідальності та роль кожного учасника в певному бізнес-процесі.

### **11. Бізнес-інтеграція**

SOA є основою для вирішення комплексного завдання бізнес-інтеграції, що полягає в організації взаємодії та управлінні бізнес-операціями, інформаційними активами організації, зовнішніми учасниками, а також в підтримці єдиної мережі співпраці керівних осіб, які відповідають за управлінські рішення. Бізнес-інтеграція полягає в об'єднанні отриманої від прикладних застосувань та різноманітних джерел даних інформації з користувачами та процесами. Ці процеси можуть бути як внутрішніми, так і зовнішніми. Основні завдання бізнес-інтеграції: інтеграція досвіду користувачів, інтеграція інформації, інтеграція застосувань, інтеграція зовнішніх учасників та інтеграція процесів. Для вирішення конкретних завдань будують бізнес-моделі, що описують організацію одного або кількох сервісів.

Таким чином, концепцією SOA пропонується збирати з блоків-сервісів необхідне комплексне ІТ-рішення, інтегруючи їх між собою за допомогою систем класу BPM. Автоматизуючи нові бізнес-процеси, можна використовувати набір уже існуючих сервісів, тому процеси, автоматизовані за допомогою SOA, легко налаштовуються або перебудовуються відповідно зі специфічними потребами організації або відповідно до змін зовнішнього середовища. При цьому BPM-система забезпечує адаптивність та підтримку SOA

та фактично виступає як управляюча система в SOA-архітектурі, що керує викликом сервісів та потоками робіт. Інтеграційні бізнес-процеси, що автоматизовані за допомогою BPM-системи, можуть об'єднувати як процеси, підтримані набором сервісів, так і процеси-сервіси в існуючих монолітних системах. Такий підхід дає змогу легко налаштувати або змінювати бізнес-процеси відповідно до специфічних потреб організації або відповідно до змін зовнішнього середовища.

*Висновки.* Проведений аналіз свідчить про те, що в основі концепції SOA лежать принципи багаторазового використання функціональних елементів, ліквідація дублювання функціональності в різноманітних програмних комплексах та застосуваннях, уніфікація типових операційних процесів, перехід операційної моделі роботи на централізовані процеси та функціональну організацію з використанням інтеграційної платформи та шини. При цьому інформаційні компоненти можуть бути розподілені за різними вузлами мережі та можуть виступати як незалежні, слабопов'язані сервіси застосування, та такі, що можуть замінюватися.

Програмні засоби, які реалізують компоненти бізнес-процесів, оформляються як сервіси, або як програмні компоненти, що реалізують закінчену функцію надання або обробки даних, переводячи їх з одного цілісного стану в інший. SOA дає змогу інтегрувати бізнес-процеси з різних компонентів, що виконуються на різних платформах, представляти у вигляді сервісів та повторно використовувати в нових бізнес-процесах успадковані компоненти.

Незважаючи на те, що архітектура SOA ґрунтується на відкритих стандартах та підтримує концепцію незалежності бізнес-інтеграції від конкретних платформ, SOA також залежить від загальної платформи, на якій має базуватися її власна інфраструктура. Така інфраструктура повинна підтримуватися всіма учасниками процесу розробки та впровадження ІС при реалізації електронних послуг у структурі інтегрованої архітектури електронних комунікацій організації.

*Перспективи подальших досліджень.* Наступним етапом дослідження технологічних аспектів впровадження концепції SOA може бути розробка єдиних стандартизованих методів проектування ІС та концепції побудови інтегрованої архітектури електронних комунікацій із використанням сервіс-орієнтованого підходу. У поєднанні з аналізом ефективних механізмів управління та впровадження ІТ-портфелів таке дослідження може скласти основу нової стратегії розвитку інтегрованих інформаційних ресурсів органів державної влади в Україні.

#### **Список використаних джерел**

1. NET Framework Conceptual Overview. - Режим доступу : <http://msdn.microsoft.com/en-us/library/zw4w595w.aspx>



2. Технология Java EE. - Режим доступу : <http://developers.sun.ru/javaee>
3. Reference Model for Service Oriented Architecture 1.0. - Режим доступу : <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>
4. Модернізація ДПС України у контексті інноваційного розвитку : матеріали наук.-практ. семінару (Ірпінь, 23 груд. 2008 р.). - Ірпінь : Нац. ун-т ДПС України, 2009. - 286 с.
5. Модернізація ДПС України у контексті інноваційного розвитку : матеріали наук.-практ. семінару (Ірпінь, 8 верес. 2010 р.). - Ірпінь : Нац. ун-т ДПС України, 2010. - 224 с.
6. UDDI Version 3.0.2. - Режим доступу : [http://uddi.org/pubs/uddi\\_v3.htm](http://uddi.org/pubs/uddi_v3.htm)
7. SOAP Версія 1.2 Часть 0 : Учебник для начинающих. - Режим доступу : <http://www.w3.org/2002/07/soap-translation/russian/part0.html>
8. Business Process Management (BPM) software and services. - Режим доступу : <http://www-142.ibm.com/software/products/us/en/category/bpm-software>
9. OASIS Web Services Business Process Execution Language (WSBPEL) TC. - Режим доступу : [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsbpel](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel)