УДК 519.257

Р.І. Дзінько, О.І. Лісовиченко, Ю.А. Рибенок

ПІДХІД ДО РОЗРОБКИ ЗАСОБІВ ІНТЕГРАЦІЇ WEB-ОРІЄНТОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ, СТВОРЕНИХ НА БАЗІ РІЗНИХ ТЕХНОЛОГІЙ

Вступ

Як правило, при розробці web–орієнтованих інформаційних систем, що націлені на використання в мережі Internet чи в локальній мережі окремої установи, постає задача створення структурно складних систем. Такі системи складаються з компонентів, кожен з яких виконує свою роль в життєвому циклі системи, і задачі, що виконуються ними, є самостійними і можуть виступати окремими повнофункціональними системами. Для таких підсистем вже є готові рішення, які успішно експлуатуються. В такому випадку, в плані економії виробничих ресурсів розробника, доцільно використати вже готовий програмний продукт, платний або з відкритим кодом.

У випадку, коли технологія, на якій базується створення такої інформаційної системи, не співпадає з технологією, на якій базується готове рішення певної підсистеми, настає проблема інтеграції цих продуктів з метою надання користувачам зручні засоби вирішення їх задач. Як показує світовий досвід, більшість розробників вважають доцільним не зв'язувати компоненти в єдине середовище користувача, що, в свою чергу, породжує значні незручності.

Наведемо в якості прикладу інформаційну систему (рис. 1), що складається з двох компонентів:

- *web–сервісу* організації списків запланованих справ, що створений з використанням мови програмування Python, технології Zope 3 (сервер додатків) та об'єктної бази даних (БД) ZODB.
- *тематичного форуму*, на якому користувачі сервісу спілкуються на теми, пов'язані з роботою основного сайту, що працює на базі технології CGI і реалізований на мові програмування PHP.

В описаній інформаційній системі є наступні проблеми, які необхідно вирішити, щоб провести успішну інтеграцію:

- відсутність єдиної системи ведення облікових записів користувачів;
- відсутність механізму єдиного входу в систему;
- дублювання даних в БД компонентів;
- розміщення систем в різних доменах, що призводить до ускладнення синхронізації сесій та даних, що зберігаються в cookie web–браузера.

© Р.І. Дзінько, О.І. Лісовиченко, Ю.А. Рибенок, 2008

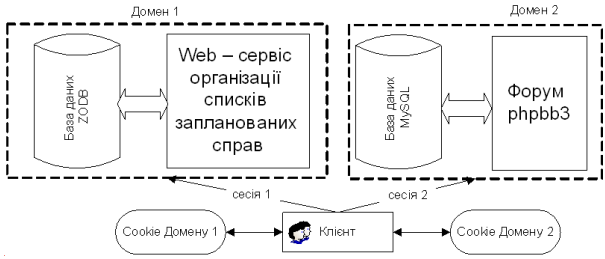

Рис. 1 – Система з двох компонентів, що потребують інтеграції

Перед постановкою задачі наведемо декілька означень:

Означення 1. *Сесія web–браузера* – сеанс роботи web - браузера з сайтом. Фактично є тимчасовим об’єктом, який унікально ідентифікує користувача і стан його взаємодії з віддаленим web–сервером [4].

Означення 2. *Cookie* – засіб довготривалого зберігання службової інформації та персональних налаштувань користувача web–системи на боці клієнта. Робота з інформацією, що зберігається в cookie ініціюється з боку сервера чи сценарію, що виконується на клієнтській частині. Дані зберігають у вигляді набору записів типу “ ключ ” : “ дані ”.

Означення 3. *Гнучкість web–орієнтованої системи* – властивість, що визначає затрати виробничих ресурсів на модифікацію, розширення чи адаптацію системи до певного середовища. Чим гнучкіша система, тим менше виробничих ресурсів потрібно затратити на виконання вищеописаних дій над системою.

Означення 4. *Захищеність системи* – властивість системи бути стійкою до спроб зловмисників неправомірними засобами заволодіти правами доступу до системи інших користувачів, в тому числі і адміністративними правами.

Означення 5. *Портативність системи* – властивість системи однакого добре працювати в різних інформаційних середовищах [4,5].

Означення 6. *Домен* – частина ієрархічного простору імен мережі Інтернет, що обслуговується групою серверів доменних імен (DNS - серверів).

Постановка задачі

Є n повнофункціональні web–орієнтовані інформаційні системи (рис. 2). Розробити загальний універсальний підхід до їх інтеграції, створених на базі різних технологій, який задовольняє наступні вимоги:

- мінімальні затрати виробничих ресурсів;
- мінімальний термін проведення інтеграції;
- максимальна захищеність кінцевої системи від вторгнень.

Реалізуємо інтеграцію в декілька етапів. Це необхідно для організації проведення інтеграції як єдиного робочого процесу, при чому на кожно-

 ISSN 1562-9945

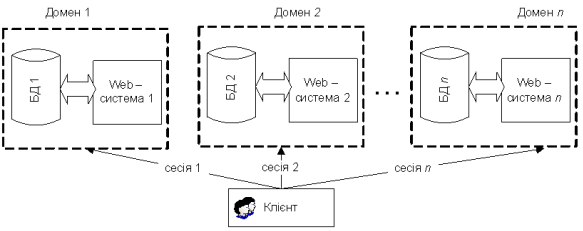

Рис. 2 – Декілька інформаційних систем, що потребують інтеграції

му наступному етапі використовуються результати, отримані в процесі виконання попередніх етапів. Отже, отримаємо наступні 5 етапів:

1. Аналіз вихідних систем на гнучкість, портативність, ресурсну ємність;
2. Адаптація компонентів до користування єдиною для них сесією web - браузера;
3. Розміщення всіх систем в єдиному домені і організація спільних cookie;
4. Оптимізація зберігання даних;
5. Адаптація систем до спільної системи авторизації та реєстрації.

Реалізація процесу інтеграції web – орієнтованих інформаційних систем

1. Аналіз вихідних систем на гнучкість, портативність, ресурсну ємність

Перший етап інтеграції передбачає аналіз систем з ціллю оптимізації використання виробничих ресурсів для створення засобів інтеграції. Кожен компонент аналізується на гнучкість, портативність та ресурсну ємність [1]. Аналіз на гнучкість та портативність проводиться для визначення компонент, які потрібно адаптувати, щоб отримати якомога менші затрати виробничих ресурсів [3]. Далі проводиться аналіз структур БД та модулів програмного забезпечення для визначення необхідних змін та доповнень, отримання параметрів апаратних ресурсів, необхідних для створення розподіленої системи [2].

У випадку системи, що наведена на рис. 1, можна в загальних рисах провести наступний аналіз. *Web–сервіс організації списків запланованих справ* (далі “ сервіс ”) створений на базі технології Zope 3, тому є інтерфейс – орієнтованою системою компонентної архітектури, що надає нам можливість змінювати функціональні властивості окремих компонентів, при цьому не впливаючи на роботу всієї системи. Мова програмування Python, за допомогою якої створений сервіс, має засоби роботи як з об’єктною БД ZODB, так і з реляційною MySQL. Наведені дані свідчать про

високу гнучкість сервісу як web-системи. Zope 3 є web–сервером, що працює на операційних системах сімейств Windows NT, Unix та Linux, що свідчить про його портативність.

Форум створений за допомогою мови програмування PHP і складається фактично з набору шаблонів web–сторінок, пов'язаних функціонально. Система важко піддається модифікації, оскільки модифікація окремих компонентів призведе до порушення правильного функціонування всього форуму, що веде за собою велику затрату виробничих ресурсів. Окрім того, мова PHP не має засобів взаємодії з об'єктною БД ZODB. Отже, форум є системою негнучкою, але портативною, оскільки працює на основі технології CGI.

На основі приведених даних можна зробити висновок, що модифікації слід розпочати з сервісу, щоб адаптувати його для взаємодії з форумом. Такий підхід дозволить значно скоротити затрати виробничих ресурсів і робочого часу.

2. Розробка засобів організації єдиної для всіх систем сесії web - браузера

Розглянемо сесію web-браузера як тимчасове місце для зберігання даних, з унікальним ідентифікатором та налаштувань системи для кожного окремого користувача. Є два способи збереження цього унікального ідентифікатора: Перший – в рядку запиту web-браузера. Такий рядок передається web-серверу при зверненні до нього методом "get" (метод звернення до web-сервера з ціллю отримання результату обробки інформації на боці сервера з заданими параметрами в рядку запиту). Другий – за допомогою cookie web-браузера.

Проте перший спосіб не завжди є зручним, адже на кожному етапі роботи потрібно відслідковувати наявність ідентифікатора сесії в рядку запиту до сервера. Використання такого підходу сильно знижує читабельність посилань, тому рекомендується користуватись механізмом cookie.

Для розробки засобів адаптації повинні бути використані результати досліджень попереднього етапу. Необхідно змінити ідентифікацію клієнта в обох системах так, щоб вони працювали зі спільним ідентифікатором сесії і зберігали цей ідентифікатор в cookie з однаковим ключом. Таким чином, при старті сесії в одній із систем, інші системи повинні бути проінформовані про те, що сесія розпочалась і для них.

В рамках розглянутого прикладу двох-компонентної системи, можна провести наступні дії. В форумі phpbb 3 сесія зберігається в cookie під ключем, значення якого зберігається. На сервері ця інформація тимчасово зберігається в БД MySQL. Якщо відповідне значення cookie з налаштованим ключем не встановлене, то система встановлює значення, згенероване за певним алгоритмом, суттю якого є послідовність hash-функцій, що застосовуються до певного текстового рядка (це сприяє підвищенню ймовірності, що ключ буде унікальним для кожного з клієнтів). Генерація ідентифікаторів сесії Zope 3 здійснюється *"менеджером ідентифіка-*

 ISSN 1562-9945

ції клієнта через cookie”, який є об’єктом, що володіє двома методами. Тому доцільніше замінити цей об’єкт іншим, що взаємодіє з БД MySQL для отримання наступної інформації:

- домену, на якому розміщено форум (приймаємо за початкові умови, що системи розміщені в різних доменах);
- ключа для зберігання ідентифікатора сесії в cookie.

Також даний об’єкт генерує унікальні ключі сесії користувачів сервісу. В силу складності алгоритму phpbb 3 для генерації ідентифікаторів сесії, спроба відтворити його на мові Python призведе до великих затрат часу. В зв’язку з цим пропонується вичленення php–скрипта, що виконує генерацію такого ідентифікатора, і повертає його клієнту при виконанні запиту до сервера.

Менеджер ідентифікації клієнтів через cookie створює також відповідний запис в таблиці сесій БД MySQL. Таким чином, незалежно від того, який із компонентів розпочав сесію, інший буде використовувати такий же ідентифікатор (тільки при умові, що будуть встановлені правильні cookie). Оскільки системи розміщені в різних доменах, то відповідні значення cookie повинні бути продубльовані для обох підсистем. Таким чином з системи, що зображена на рис. 1 ми отримуємо наступну систему (рис. 3).

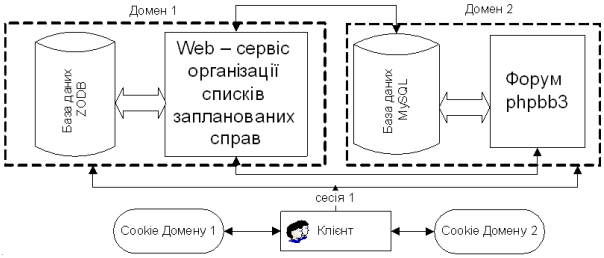

Рис. 3 – Взаємозв‘язок інформаційних систем при організації спільної сесії web-браузерів

3. Розміщення всіх підсистем систем в єдиному домені і організація спільних cookie

Умова розміщення всіх підсистем в єдиному домені є ключовим моментом організації зручності роботи користувача, а також спрощує процес організації спільних cookie, оскільки, як стало зрозуміло на попередньому етапі, їх не потрібно буде дублювати. Такий підхід: по-перше, спрощує контроль над cookie; по-друге, підвищує надійність роботи; по-третє, знижує ймовірність виникнення помилок на етапі програмування. Якщо ж не дотримуватись такої умови, то інформацію кожного користувача, що призначена для збереження, необхідно дублювати для кожного домену з системи доменів, в межах спроектованої інформаційної системи. Завдяки

розміщенню підсистем в єдиному домені, розробка менеджера ідентифікації клієнта через cookie спрощується, адже зникає необхідність дублювання даних cookie для двох доменів. З'являється можливість створення єдиного механізму авторизації клієнтів системи через cookie.

Дану систему доцільно розмістити на одному сервері під управлінням web – сервера, що задовольняє наступні вимоги: робота з CGI – інтерфейсом; робота з WSGI – інтерфейсом; підтримка правил заміни посилань за допомогою регулярних виразів (символьні вирази, що дозволяють задати шаблон текстового рядка).

Для таких цілей ідеально підходить сервер з відкритим кодом Apache 2.x.x., завдяки якій ми можемо розмістити всю систему в наступному вигляді:

- сервіс – http://example.com/
- форум – http://example.com/forum

Така організація (рис. 4) дозволяє вирішити наступні проблеми: проблему спільних cookie; спрощує посилання на інформаційну систему; підвищує зручність роботи клієнтів; надає змогу створити єдину систему авторизації клієнтів, що означає наступне: коли клієнт авторизувався в одній із підсистем, він стає автоматично авторизованим і у всіх інших [1].

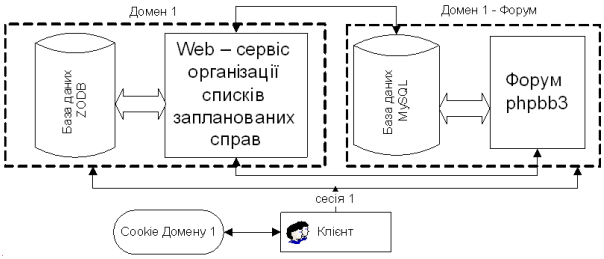

Рис. 4 – Взаємозв'язок інформаційних систем при організації спільної сесії web-браузерів, розміщених на одному web – сервері

4. Оптимізація зберігання даних компонентів інформаційної системи в БД

В більшості випадків кожен компонент системи використовує свою БД, які можуть базуватись на різних СУБД. Якщо це БД одного типу (наприклад, всі БД є реляційними), то задача оптимізації зберігання даних може звестись до перенесення всіх БД на єдину СУБД, що передбачає невелику затрату виробничих ресурсів та вимог до апаратного забезпечення.

Більш проблематичною є ситуація, коли використовувані БД є різнотипними (наприклад, використовуються об'єктні і реляційні БД). Для таких випадків потрібно застосувати персональний підхід в кожній

окремій ситуації, враховуючи те, що деякі одних підсистем можуть знадобитися для роботи інших. Тому результатом роботи даного етапу повинна бути СУБД з мінімальним дублюванням даних. У випадку розглянутої системи, можна зробити наступні висновки:

- дублювання даних відбувається за рахунок збереження інформації про користувачів і в ZODB і в БД MySQL;
- дані, що зберігаються про користувачів в БД сервісу (ZODB) є підмножиною даних, що зберігаються в БД форуму (MySQL).

Тому доцільно модифікувати сервіс так, щоб він використовував дані з БД MySQL. Клієнти сервісу зберігаються в ZODB і існують в межах системи Zope 3 у вигляді об'єктів, тоді як клієнти форуму представлені у вигляді запису в реляційній таблиці. Для реалізації користувачів сервісу на базі БД MySQL необхідно перевизначити клас користувача. Для цього можна використати бібліотеку Python SQL Alchemy, яка, по своїй суті є ORM (object relational mapper – засіб, що використовує технологію для створення об'єктів на основі даних з реляційних таблиць). Таким чином методи класу клієнта залишаться такими ж, а поля даних класу отримуватимуть інформацію не з ZODB, а з БД MySQL [4,5].

Повне злиття двох БД компонентів для використання з єдиною СУБД в випадку даної двокомпонентної системи недоцільне, адже спричиняє багато труднощів, що пов'язані з різним способом організації даних в цих БД. Спроба проведення такого злиття призведе до значних модифікацій в програмному коді сервісу.

5. Адаптація систем до спільної системи авторизації та реєстрації

Слідування попереднім етапам інтеграції дозволяє вирішити наступні проблеми:

- розробити єдиний для всієї інформаційної системи (і для всіх її компонентів) метод авторизації та реєстрації;
- забезпечує єдине місце зберігання інформації про дані кожного користувача, тому вирішується проблема створення єдиного облікового запису;
- забезпечує безперешкодне пересування користувача всередині системи без проміжних авторизації для доступу до функцій кожного окремого компоненту.

Таким чином, основною задачею даного пункту є створення єдиного механізму реєстрації користувача.

В межах двох компонентної системи поставлена задача вирішується із встановленням додаткових відомостей про клієнта в cookie - виконується розширення вищеописаного менеджера ідентифікації клієнта через cookie. Він встановлює ідентифікатор користувача з БД MySQL, а також прапорець автоматичного входу в систему при умові закриття старої сесії користувача і початку нової.

Що стосується реєстрації користувача, доцільно залишити процес реєстрації користувачів таким, як він є для форуму і для сервісу, оскільки створені на цьому і попередніх етапах модифікації, забезпечують єдину систему авторизації і збереження даних про користувачів. Можливим також є варіант створення єдиної форми реєстрації чи використання однієї з двох наявних форм реєстрації сервісу і форуму. В другому випадку потрібно організувати механізм повернення на попередній етап роботи з системою після реєстрації.

Висновки

Запропонований підхід до інтеграції web-орієнтованих інформаційних систем вирішує актуальні задачі, що постають в процесі інтеграції систем, створених на базі різних технологій.

Описаний підхід дозволяє:

- мінімізувати дублювання інформації в БД;
- зменшити вимоги до апаратних ресурсів
- вирішити деякі принципові незручності, які можуть виникнути в процесі користування, завдяки створенню єдиної системи авторизації та реєстрації, розміщенню компонентів в єдиному домені та встановленню правил заміни посилань.
- зменшити затрати виробничих ресурсів на створення такої системи, створювати програмний код та скрипти для багаторазового використання.
- підвищити надійність системи завдяки існуванню єдиної системи ведення облікових записів.

Особливу увагу необхідно звернути на захищеність інтегрованої інформаційної системи, що є в багатьох випадках ключовим питанням, особливо коли така система є комерційним сервісом. В цьому випадку дані про облікові записи користувачів повинні бути особливо захищеними. Безперечно захищеність такої системи визначається рівнем захисту найбільш вразливого до атак зловмисників компонента. Тому, коли настає вибір створення власних компонентів системи чи використання вже готових, слід спочатку оцінити доцільність того чи іншого вибору. У випадку розглянутої двокомпонентної системи, захищеність форуму як вибраного вже готового компоненту не важлива, оскільки дані з БД форуму не є закритою інформацією. Боротьба з можливістю втрати інформації вирішується періодичним резервним копіюванням вмісту БД. Проте конфіденційна інформація міститься в БД ZODB сервісу. СУБД ZODB не дозволяє отримати віддалений доступ до інформації, яка зберігається в ній, без встановлених відповідних програмних засобів, і заволодіти інформацією можна тільки шляхом отримання адміністративних прав на сервері чи виконанням Python – програми на стороні сервера.

 ISSN 1562-9945

Література

1. Web services: problems and future directions - Hongbing Wang, Joshua Zhexue Huang, Yuzhong Qu and Junyuan Xie, April 2004, http://www.sciencedirect.com.
2. Component Oriented Programming - Clemens Szyperski, Jan Bosch, Wolfgang Weck., Springer publishing, 1999.
3. A component based services architecture for building distributed applications Bramley, R. Chiu, K. Diwan, S. Gannon, D. Govindaraju, M. Mukhi, N. Temko, B. Yechuri, M. 2000. Department. of Computer. Science., Indiana University., Bloomington, Indiana, USA.
4. Документація для розробників і адміністраторів фреймворку для розробки web – орієнтованих інформаційних систем Zope 3. http://www.zope.org/Documentation/
5. Документація для користувачів, розробників та адміністраторів СУБД MySQL. http://dev.mysql.com/doc/

Получено 17.11.2008